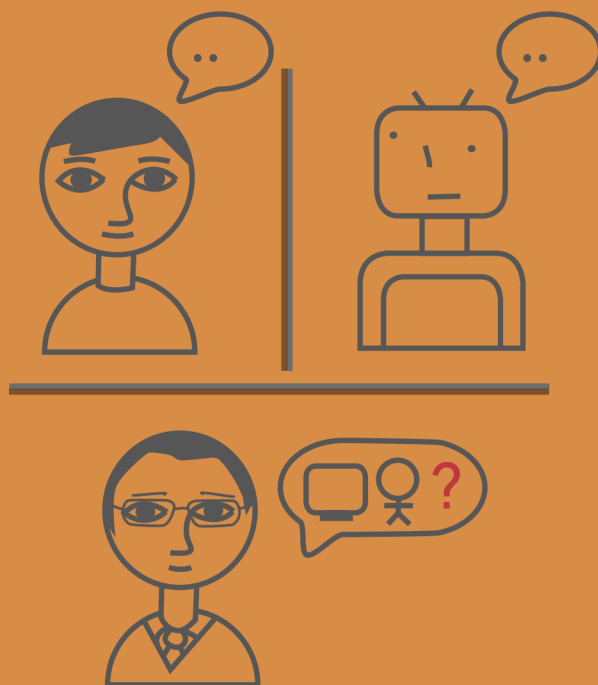


ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Μια Εισαγωγική Προσέγγιση

ΚΑΤΕΡΙΝΑ ΓΕΩΡΓΟΥΛΗ



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALLINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ανάπτυξη στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ



ΕΣΠΑ
2007-2013
ανάπτυξη στην κοινωνία της γνώσης
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τεχνητή Νοημοσύνη

Συγγραφή

Κατερίνα Γεωργούλη

Κριτικός αναγνώστης

Σωκράτης Κάτσικας

Συντελεστές έκδοσης

ΓΡΑΦΙΣΤΙΚΗ ΕΠΙΜΕΛΕΙΑ: **Αναστασία Αλεξιάκου**

ΤΕΧΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ: **Σοφία Λουκέρη**

Copyright © ΣΕΑΒ, 2015



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 3.0.

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN: 978-960-603-031-4

Στους φοιτητές μου για τα δημιουργικά χρόνια που μου προσέφεραν.

Πίνακας περιεχομένων

Πίνακας περιεχομένων.....	4
Πίνακας συντομεύσεων-ακρωνύμια	10
Πρόλογος.....	11
Εισαγωγή	12
Τεστ Τούρινγκ.....	12
Ορισμός TN	13
Υπολογιστική Νοημοσύνη.....	15
Ιστορική Εξέλιξη της Τεχνητής Νοημοσύνης.....	16
Η έρευνα στο χώρο της TN	17
Τεχνητή και Περιρρέουσα Νοημοσύνη	20
Σύγχρονες εφαρμογές της TN	21
Βιβλιογραφία/Αναφορές	22
ΚΕΦΑΛΑΙΟ 1 - Επίλυση Προβλημάτων.....	24
1.1 Προβλήματα και αναζήτηση λύσης.....	24
1.1.1 Τύποι προβλημάτων.....	24
1.1.2 Προβλήματα που επιλύονται με αναζήτηση	25
1.1.3 Ορισμός Προβλήματος	26
1.1.4 Παραδείγματα προβλημάτων.....	28
Προβλήματα-παιχνίδια	28
Προβλήματα πραγματικού κόσμου.....	34
1.1.5 Λύση προβλήματος.....	34
1.2 Αναζήτηση λύσης.....	36
1.2.1 Δένδρο Αναζήτησης	36
1.2.2 Περιγραφή Δένδρων Αναζήτησης.....	39
1.3 Αλγόριθμοι Αναζήτησης.....	41
1.3.1 Γενικός αλγόριθμος αναζήτησης.....	42
1.4 Τυφλές μέθοδοι αναζήτησης	45
1.4.1. Πρώτα σε Βάθος Αναζήτηση.....	45
1.4.2 Η Πρώτα σε Πλάτος Αναζήτηση	48
1.4.3 Αλγόριθμος Αναζήτησης Επαναληπτικής Εμβάθυνσης.....	51
1.4.4 Μέθοδος Αναζήτησης Διπλής Κατεύθυνσης.....	52
1.5 Ευρετικές μέθοδοι αναζήτησης	53
1.5.1 Ευρετικές Συναρτήσεις.....	54
1.5.2 Αναζήτηση Καλύτερης-Πρώτης.....	55
1.5.3. Αναζήτηση με αναρρίχηση λόφων	57

1.5.4 Αναζήτηση Βασικής Επέκτασης και Οριοθέτησης	60
1.5.5 Ακτινωτή μέθοδος αναζήτησης.....	61
1.5.6 Μέθοδος αναζήτησης Άλφα-Άστρο	63
Βιβλιογραφία/Αναφορές	66
Κριτήρια αξιολόγησης.....	67
ΚΕΦΑΛΑΙΟ 2 - Αναπαράσταση Γνώσης.....	69
2.1 Γνώση.....	69
2.1.1 Είδη Γνώσης	72
2.2 Αναπαράσταση Γνώσης.....	74
2.2.1 Χαρακτηριστικά μιας «καλής» αναπαράστασης γνώσης	74
2.2.2 Μέθοδοι Αναπαράστασης Γνώσης	75
2.2.3 Σχήματα Λογικής Αναπαράστασης	76
Προτασιακός Λογισμός.....	76
Κατηγορηματικός Λογισμός.....	77
Κατηγορηματικός Λογισμός Πρώτης Τάξης	78
2.2.4 Δομημένες Μορφές Αναπαράστασης Γνώσης	80
Σημασιολογικά Δίκτυα	80
Τα σημασιολογικά δίκτυα ως εννοιολογικοί χάρτες.....	85
Σχήματα	88
Πλαίσια	88
Σενάρια.....	92
Κανόνες	94
Οντολογίες	96
Βιβλιογραφία/Αναφορές	99
Κριτήρια αξιολόγησης.....	100
ΚΕΦΑΛΑΙΟ 3 - Συστήματα Κανόνων.....	102
3.1 Συλλογιστική.....	102
3.1.1 Επαγωγή	102
3.1.2 Απαγωγή	103
3.1.3 Παραγωγή.....	104
3.2 Συστήματα Κανόνων	105
3.2.1 Υλοποίηση κανόνων.....	105
3.2.2 Περιγραφή Συστημάτων Κανόνων	107
3.3 Συστήματα Εξαγωγής Συμπερασμάτων	108
3.4 Συστήματα παραγωγής	110
3.4.1 Κύκλοι Λειτουργίας	111
3.4.2 Δομή Ελέγχου	112

3.4.3 Προς τα εμπρός αλυσιδωτή εκτέλεση κανόνων	113
3.4.4 Προς τα πίσω αλυσιδωτή εκτέλεση κανόνων	119
3.4.5 Ο Γενικός Επιλυτής Προβλημάτων (General Problem Solver)	123
Βιβλιογραφία/Αναφορές	124
Κριτήρια αξιολόγησης.....	125
ΚΕΦΑΛΑΙΟ 4 –Μηχανική Μάθηση.....	127
4.1 Εισαγωγή	127
4.2 Είδη Μηχανικής Μάθησης	128
4.3 Αλγόριθμοι Επιβλεπόμενης Επαγωγικής Μάθησης.....	130
4.3.1 Δένδρα Απόφασης	131
Αλγόριθμος δένδρου απόφασης.....	134
Ο αλγόριθμος ID3.....	135
Επιλογή χαρακτηριστικού	136
Παράδειγμα ID3	139
Παράδειγμα δημιουργίας δένδρου απόφασης με χρήση εντροπίας και κέρδους πληροφορίας	139
Άλλα παραδείγματα εφαρμογής ID3	143
Από δένδρα απόφασης σε κανόνες	147
Πλεονεκτήματα και Μειονεκτήματα των ΔΑ	148
4.4 Μάθηση με Συλλογιστική Βασισμένη σε Περιπτώσεις	149
4.4.1 Χαρακτηριστικά της ΣΒΠ/CBR	150
4.4.2 Ο Κύκλος ΣΒΠ/CBR.....	150
4.4.3 Κύριοι τύποι μεθόδων ΣΒΠ/CBR.....	152
4.4.4 Πλεονεκτήματα - Μειονεκτήματα	153
4.4.5 Δυσχέρεια στην απόκτηση γνώσης	154
4.5 Νευρωνικά Δίκτυα	155
4.5.1 Γενικά.....	155
4.5.2 Προσομοίωση φυσικών νευρωνικών δικτύων με τεχνητά νευρωνικά δίκτυα	157
4.5.3 Ιστορική Αναδρομή.....	159
4.5.4 Αρχιτεκτονική ΤΝΔ	159
Πολυεπίπεδα ΤΝΔ	161
Συναρτήσεις μετάβασης	161
4.5.5 ΤΝΔ Πρόσθιας Τροφοδότησης	163
Perceptrons	163
Παράδειγμα λειτουργίας πολυεπίπεδου ΤΝΔ για το xor.....	164
Στάδια ολοκλήρωσης ΤΝΔ.....	165
4.5.6 Εκπαίδευση ΤΝΔ.....	166
Εκπαίδευση Perceptron.....	167
Κανόνες Εκμάθησης.....	168

Σφάλματα.....	169
Εκπαίδευση ΤΝΔ με τη μέθοδο οπισθοδιάδοσης σφάλματος.....	169
Αλλαγές στα βάρη	170
Μη Επιβλεπόμενη Μάθηση ΤΝΔ – Δίκτυα Kohonen	171
Παράδειγμα εκπαίδευσης δικτύου Kohonen	172
Ανατροφοδοτούμενα ΤΝΔ - Δίκτυα Hopfield	174
4.5.7 Προτερήματα ΤΝΔ.....	175
4.5.8 ΤΝΔ και Ανθρώπινος Εγκέφαλος	176
4.5.9 Εφαρμογές ΤΝΔ	177
4.6 Γενετικοί Αλγόριθμοι	178
4.6.1 Φυσική εξέλιξη: Η αρχική έμπνευση	178
4.6.2 Ιστορική Εξέλιξη.....	179
4.6.3 Γενικά περί Γενετικών Αλγορίθμων	180
4.6.4 Η Ανατομία ενός Γενετικού Αλγορίθμου	181
4.6.5 Κωδικοποίηση.....	182
4.6.6 Η αρχικοποίηση ενός πληθυσμού.....	183
4.6.7 Θεωρία σχημάτων	184
4.6.8 Αναπαραγωγή	185
Γονική επιλογή	185
4.6.9 Γενετικοί Τελεστές.....	189
4.6.10 Διασταύρωση.....	190
Διασταύρωση Ενός Σημείου.....	190
Διασταύρωση Πολλαπλών Σημείων	191
Ομοιόμορφη Διασταύρωση	191
4.6.11 Μετάλλαξη (Mutation)	192
4.6.12 Γενεαλογική Αντικατάσταση	194
4.6.13 Εκλεκτικές Στρατηγικές.....	195
4.6.14 Ολοκλήρωση ΓΑ	195
4.6.15 Διαφορές από άλλες τεχνικές.....	196
4.6.16 Εφαρμογές Γενετικών Αλγορίθμων.....	196
Βιβλιογραφία/Αναφορές	197
Κριτήρια αξιολόγησης.....	198
ΚΕΦΑΛΑΙΟ 5 - Έμπειρα Συστήματα	200
5.1 Γενικά.....	200
5.2 Η Δομή ενός ΕΣ Βασισμένου σε Κανόνες.....	201
5.3 Χαρακτηριστικά ενός ΕΣ	202
5.4 Ανάπτυξη ενός ΕΣ.....	204
5.5 Απόκτηση Γνώσης	205

5.5.1 Συγκέντρωση Γνώσης.....	205
5.6 Αβεβαιότητα.....	206
5.6.1 Συντελεστές Βεβαιότητας.....	207
5.7 Ασάφεια	209
5.7.1 Ασαφείς κανόνες.....	211
5.8 Προτερήματα και Μειονεκτήματα Εμπείρων Συστημάτων Βασισμένων σε Κανόνες	214
Βιβλιογραφία/Αναφορές	215
Κριτήρια αξιολόγησης.....	215
ΚΕΦΑΛΑΙΟ 6 - Νοήμονες Πράκτορες	217
6.1 Εισαγωγικά.....	217
6.2 Ορισμοί πρακτόρων.....	218
6.3 Κοινά Χαρακτηριστικά Πρακτόρων	220
6.4 Κατηγορίες Πρακτόρων	221
6.4.1 Ορθολογικοί πράκτορες.....	223
6.5 Σύγχρονες Θεωρήσεις Πρακτόρων.....	223
6.6 Περιβάλλοντα Πρακτόρων.....	224
6.7 Αφηρημένες Αρχιτεκτονικές Πρακτόρων.....	225
6.7.1 Αντιδραστικοί Πράκτορες.....	226
6.7.2 Πράκτορες με Εσωτερική Κατάσταση	227
Πράκτορες με στόχους	228
Πράκτορες με Πεποιθήσεις– Επιθυμίες– Προθέσεις.....	229
Πράκτορες που μαθαίνουν.....	231
6.7.3 Υβριδικοί Πράκτορες	232
6.7.4 Κινητοί Πράκτορες	233
6.8 Πολυπρακτορικά Συστήματα	233
6.8.1 Χαρακτηριστικά πολυπρακτορικών συστημάτων	234
6.8.2 Κρίσιμα σημεία στη σχεδίαση & υλοποίηση	234
6.8.3 Μοντέλα διασύνδεσης πολυπρακτορικών συστημάτων.....	235
6.8.4 Επικοινωνία μεταξύ πρακτόρων.....	237
6.8.5 Αλληλεπίδραση Πρακτόρων	237
6.8.6 Πολυπρακτορικό περιβάλλον	238
6.8.7 Γλώσσες επικοινωνίας	238
6.9 Εφαρμογές πρακτόρων	239
Βιβλιογραφία/Αναφορές	239
Κριτήρια αξιολόγησης.....	240
ΚΕΦΑΛΑΙΟ 7 - Επεξεργασία και Κατανόηση Φυσικής Γλώσσας.....	242
7.1 Εισαγωγή	242
7.2 Δυσκολίες στην Κατανόηση Φυσικής Γλώσσας.....	242

7.3 Ιστορική Αναδρομή	243
7.4 Πώς πραγματοποιείται η Επεξεργασία Φυσικής Γλώσσας σήμερα	246
7.4.1 Λεξικό.....	246
7.5 Συντακτική Ανάλυση.....	247
7.6 Γραμματική	247
7.6.1 Γραμματικές Οριστικών Προτάσεων.....	248
7.6.2 Συντακτικό Δένδρο	249
7.6.3 Διαγράμματα Μετάβασης.....	250
7.7 Αλγόριθμοι Διόρθωσης Ορθογραφικών Λαθών	252
7.7.1 Μοντέλο Χαρακτήρων Διόρθωσης Λαθών	252
7.7.2 Ηχητικό Μοντέλο Διόρθωσης Λαθών	253
7.8 Σημασιολογική ανάλυση	253
7.8.1 Αντιμετώπιση αμφιβολίας στην ερμηνεία.....	253
7.9 Πραγματολογική ανάλυση.....	254
7.10 Παραγωγή Φυσικής Γλώσσας	255
7.11 Αναγνώριση ομιλίας	255
7.12 Πεδία εφαρμογής Επεξεργασίας Φυσικής Γλώσσας	256
Βιβλιογραφία/Αναφορές	256
Κριτήρια αξιολόγησης.....	257
Ευρετήριο Όρων	259

Πίνακας συντομεύσεων-ακρωνύμια

AI	ARTIFICIAL INTELLIGENCE
AmI	AMBIENT INTELLIGENCE
ANN	ARTIFICIAL NEURAL NETWORKS
BB	BRANCH AND BOUND SEARCH
BDI	AGENTS WITH BELIEFS-DESIRES-INTENTIONS
BestFS	BEST FIRST SEARCH
BFS	BREADTH FIRST SEARCH
CBR	CASE-BASED REASONING
CI	COMPUTATIONAL INTELLIGENCE
DFS	DEPTH FIRST SEARCH
DGG	DEFINITE CLAUSE GRAMMARS
EHC	EINFORCED HILL CLIMBING
ES	EXPERT SYSTEMS
FIPA	FOUNDATION FOR INTELLIGENT PFYSICAL AGENTS
GA	GENITIC ALGORITHMS
GPS	GENERAL PROBLEM SOLVER
HC	HILL CLIMBING SEARCH
IDS	ITERATIVE DEEPENING SEARCH
KBS	KNOWLEDGE BASED SYSTEMS
NLP	NATURAL LANGUAGE PROCESSING
NLU	NATURAL LANGUAGE UNDERSTANDING
RBS	RULE-BASED SYSTEMS-
B&B	BASIC BRANCH AND BOUND SEARCH
ΓΑ	ΓΕΝΕΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ
ΓΕΠ	ΓΕΝΙΚΟΣ ΕΠΙΛΥΤΗΣ ΠΡΟΒΛΗΜΑΤΩΝ
ΓΟΠ	ΓΡΑΜΜΑΤΙΚΕΣ ΟΡΙΣΤΙΚΩΝ ΠΡΟΤΑΣΕΩΝ
ΔΑ	ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ
ΕΣ	ΈΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ
ΕΧ	ΕΝΝΟΙΟΛΟΓΙΚΟΣ ΧΑΡΤΗΣ
ΠΕΠ	ΠΡΑΚΤΟΡΕΣ ΜΕ ΠΕΠΟΙΘΗΣΕΙΣ-ΕΠΙΘΥΜΙΕΣ-ΠΡΟΘΕΣΕΙΣ
ΣΒΓ	ΣΥΣΤΗΜΑΤΑ ΒΑΣΙΣΜΕΝΑ ΣΤΗ ΓΝΩΣΗ
ΣΒΚ	ΣΥΣΤΗΜΑΤΑ ΒΑΣΙΣΜΕΝΑ ΣΕ ΚΑΝΟΝΕΣ
ΣΒΠ	ΣΥΛΛΟΓΙΣΤΙΚΗ ΒΑΣΙΣΜΕΝΑ ΣΕ ΠΕΡΙΠΤΩΣΕΙΣ
ΤΝΔ	ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ
ΤΝ	ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
ΥΝ	ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

Πρόλογος

Η επιστημονική περιοχή – κατά άλλους ξεχωριστή επιστήμη- της Τεχνητής Νοημοσύνης πρωτοεμφανίστηκε ως διακριτός κλάδος ερευνητικής αναζήτησης τη δεκαετία του 1950. Σήμερα, έχοντας διανύσει μεγάλη απόσταση, έχοντας παράξει σημαντικότερα αποτελέσματα που έχουν συντελέσει καθοριστικά στην τεχνολογική πρόοδο αλλά και στην επαύξηση της γνώσης μας και του βαθμού κατανόησής μας για το πώς λειτουργεί ο κόσμος, αποτελεί διδακτικό αντικείμενο κορμού σε όλα τα προγράμματα σπουδών που πραγματεύονται, με τον ένα ή τον άλλο τρόπο, την επιστήμη της Πληροφορικής. Δεν είναι, όμως, σπάνιο το φαινόμενο να διδάσκεται και σε προγράμματα σπουδών εντελώς διαφορετικών επιστημών, όπως π.χ. γνωσιακών σπουδών, σπουδών ψυχολογίας κ.ά.

Είναι, επομένως, αυτονόητη η ανάγκη συμπερίληψης στο σύνολο των ηλεκτρονικών βιβλίων που αναπτύσσονται στο πλαίσιο της δράσης «ΚΑΛΛΙΠΟΣ» και ενός βιβλίου με αντικείμενο την Τεχνητή Νοημοσύνη. Το βιβλίο αυτό, γραμμένο από την καλή συνάδελφο και φίλη Κατερίνα Γεωργούλη, Καθηγήτρια του Τμήματος Πληροφορικής του Τ.Ε.Ι. Αθήνας, κρατάτε στα χέρια σας.

Το βιβλίο είναι δομημένο σε επτά κεφάλαια. Στην αρχή, όπως συμβαίνει με σχεδόν κάθε βιβλίο που προορίζεται για να υποστηρίξει την εκπαιδευτική διαδικασία, υπάρχει μια εισαγωγή στα κύρια θέματα και στις βασικές έννοιες της τεχνητής νοημοσύνης. Το πρώτο κεφάλαιο αναφέρεται στα προβλήματα και στην επίλυσή τους με μεθόδους αναζήτησης, με έμφαση στην αναζήτηση με χώρο καταστάσεων. Το δεύτερο κεφάλαιο είναι αφιερωμένο σε ένα κεντρικό μέρος της επιστημονικής περιοχής της τεχνητής νοημοσύνης, την αναπαράσταση γνώσης. Στο κεφάλαιο αυτό αναλύεται η έννοια της γνώσης και παρουσιάζονται οι βασικές λογικές και δομημένες μέθοδοι αναπαράστασης γνώσης. Το τρίτο κεφάλαιο αναφέρεται στα συστήματα κανόνων και ειδικότερα στα συστήματα παραγωγής. Το τέταρτο κεφάλαιο στρέφει την προσοχή μας σε ένα άλλο κεντρικό αντικείμενο της τεχνητής νοημοσύνης, τη μηχανική μάθηση και ιδιαίτερα την επαγωγική μάθηση. Το πέμπτο κεφάλαιο παρουσιάζει τον βασικό εκπρόσωπο των συστημάτων βασισμένων σε γνώση, δηλαδή τα έμπειρα συστήματα βασισμένα σε κανόνες. Στο έκτο κεφάλαιο συζητούνται οι βασικές αφηρημένες αρχιτεκτονικές των νοημόνων πρακτόρων, γίνεται αναφορά στα πολυπρακτορικά συστήματα και παρουσιάζονται εφαρμογές τους. Το βιβλίο κλείνει με το έβδομο και τελευταίο κεφάλαιο, το οποίο αναφέρεται σε μια από τις δημοφιλέστερες –αν όχι τη δημοφιλέστερη- και γνωστότερες στο ευρύ κοινό εφαρμογές της τεχνητής νοημοσύνης, την επεξεργασία και κατανόηση φυσικής γλώσσας.

Το αντικείμενο της Τεχνητής Νοημοσύνης δεν εξαντλείται με τα παραπάνω. Όμως, τα περιεχόμενα του βιβλίου αυτού, που έχει ως κύριο στόχο να υποστηρίξει την εκπαιδευτική διαδικασία στην τριτοβάθμια εκπαίδευση στην Ελλάδα, καλύπτουν απολύτως το αντικείμενο σε πρώτο, εισαγωγικό επίπεδο και σε έκταση τέτοια που μπορεί να καλυφθεί διδακτικά σε δεκατρείς τριώρες εβδομαδιαίες διαλέξεις, όσο δηλαδή, θεωρητικά τουλάχιστον, διαρκεί ένα ακαδημαϊκό εξάμηνο στα ελληνικά ΑΕΙ. Έτσι, το βιβλίο αποτελεί εξαιρετική επιλογή ως διδακτικό βοήθημα σε προπτυχιακό αλλά και εισαγωγικό μεταπτυχιακό επίπεδο.

Απήλαυσα την κριτική ανάγνωση του κειμένου που η Κατερίνα είχε την καλοσύνη να μου εμπιστευθεί. Ελπίζω να ανταποκρίθηκα στις προσδοκίες της και, ακόμη περισσότερο, ελπίζω να απολαύσετε και σεις τη μελέτη του βιβλίου και, μέσω αυτής, να αγαπήσετε το αντικείμενο της τεχνητής νοημοσύνης. Ευνόητο είναι ότι κάθε καλόπιστη κριτική, θετική ή αρνητική, είναι ευπρόσδεκτη, όπως άλλωστε ορίζει η ακαδημαϊκή δεοντολογία.

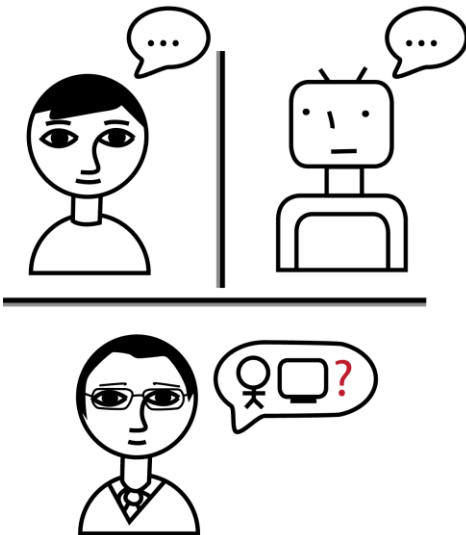
Σωκράτης Κ. Κάτσικας
Αθήνα, Σεπτέμβριος 2015

Εισαγωγή

Αν και η **Τεχνητή Νοημοσύνη-TN** (Artificial Intelligence - AI) τυπικά διατυπώθηκε ως όρος για πρώτη φορά στο Dartmouth College το 1956, σε συνέδριο ερευνητών από τους χώρους των Μαθηματικών, της Ηλεκτρονικής και της Ψυχολογίας για τη μελέτη δυνατοτήτων χρήσης των υπολογιστών ως προς την προσομοίωση της ανθρώπινης νοημοσύνης (McCarthy, Allen Newell, Herbert Simon και Marvin Minsky), στην πραγματικότητα ως έννοια είχε ήδη κάνει την εμφάνισή της το 1950, σε μελέτη του Άλαν Τούρινγκ (1912-1954), στην οποία ο φημισμένος Άγγλος μαθηματικός έθετε το ερώτημα: «Μπορούν οι μηχανές να σκεφτούν;». Βεβαίως, το ερώτημα του Τούρινγκ, το οποίο ακόμη δεν μπορεί να απαντηθεί με βεβαιότητα, και εν γένει η μελέτη της ανθρώπινης νοημοσύνης, απασχολούν τον άνθρωπο ήδη από τις απαρχές της ιστορίας τόσο στη φιλοσοφία με κυριότερο εκπρόσωπο τον Καρτέσιο (1596–1650) όσο και στη λογοτεχνία, με καλύτερα δείγματα τον *Πυγμαλίωνα* του Τζωρτζ Μπέρναρντ Σω και τον *Δόκτορα Φρανκεστάιν* της Μαίρης Σέλλεϋ. Εξάλλου, αρκετά λογοτεχνικά κείμενα του '60 αντικατοπτρίζουν τις φιλοδοξίες των πρώτων ερευνητών της TN (λ.χ. το 2001: *Η Οδύσσεια του Διαστήματος* του Άρθουρ Κλαρκ).

Τεστ Τούρινγκ

Ο Τούρινγκ θεώρησε ότι η ερώτηση που έθεσε ήταν αρκετά αόριστη και ως εναλλακτική λύση πρότεινε ένα παιχνίδι μίμησης που έμεινε στην ιστορία της Πληροφορικής ως *τεστ Τούρινγκ (Turing test)*. Στο εν λόγω παιχνίδι παίρνουν μέρος ένας «ανακριτής», ένα φυσικό πρόσωπο και μία υπολογιστική μηχανή. Ο ανακριτής βρίσκεται σε διαφορετικό χώρο από το φυσικό πρόσωπο και τη μηχανή, τους απευθύνει μια σειρά ερωτήσεων και δέχεται τις απαντήσεις με τέτοιο τρόπο, ώστε να είναι αδύνατο να αντιληφθεί ποιος από τους άλλους δύο του απαντά κάθε φορά.



Εικόνα 1 Τεστ Τούρινγκ¹

Από τον τρόπο με τον οποίο δόθηκε η απάντηση ο ανακριτής οφείλει να συμπεράνει σωστά, αν αυτός που απάντησε ήταν το πρόσωπο ή η μηχανή. Το 1950, στην εργασία του *Computing Machinery and Intelligence*² ο Τούρινγκ έγραφε ότι έως το έτος 2000 θα υπάρχουν τόσο έξυπνες μηχανές, ώστε η πιθανότητα ο ανακριτής να κάνει λάθος, να θεωρήσει, δηλαδή, πως η απάντηση που έλαβε στην ερώτησή του προέρχεται από ένα φυσικό πρόσωπο, ενώ στην πραγματικότητα προέρχεται από μηχανή, θα είναι μεγαλύτερη του 30%.

¹ Από το άρθρο Turing, Father of the Modern Computer, <http://www.rutherfordjournal.org/article040101.html>

² <http://loebner.net/Prizef/TuringArticle.html>

Μπορεί η πρόβλεψη του Τούρινγκ να μην επαληθεύθηκε τη χρονολογία που αναφέρεται, αλλά σίγουρα μεταξύ μιας μεγάλης μερίδας ερευνητών του κλάδου της ΤΝ επικρατεί η άποψη ότι ο επιζητούμενος στόχος δεν είναι ανέφικτος, δεδομένης της προόδου της επιστήμης της Πληροφορικής.

Προς την κατεύθυνση αυτή έχουν δημιουργηθεί διαγωνισμοί βασισμένοι στο τεστ Τούρινγκ, με πιο γνωστό το *Loebner Prize in Artificial Intelligence*³.

Ο προγραμματισμός ενός υπολογιστή, ώστε να υποβληθεί σε παρόμοια τεστ, απαιτεί τη συμμετοχή αρκετών επιστημονικών πεδίων, όπως είναι η επεξεργασία φυσικής γλώσσας, η αναπαράσταση γνώσης και η αυτοματοποιημένη συλλογιστική. Μία επέκταση του τεστ (πλήρες τεστ Τούρινγκ) περιλαμβάνει και την αναγνώριση εικόνων και αντικειμένων με τη συμμετοχή και άλλων δύο επιστημονικών πεδίων, της μηχανικής όρασης (machine vision) και της ρομποτικής (robotics).



Εικόνα 2 Το chatbot Eugene Goostman που πέρασε το Τεστ Τούρινγκ το 2014 - <http://www.zdnet.com/article/computer-chatbot-eugene-goostman-passes-the-turing-test/>

Τον Ιούνιο του 2014, για πρώτη φορά ένα πρόγραμμα υπολογιστή, το Eugene Goostman, μετά από πολλές συμμετοχές σε αντίστοιχους διαγωνισμούς, πέρασε το πλήρες τεστ Τούρινγκ του 2014 που πραγματοποιήθηκε στη φημισμένη Royal Society του Λονδίνου⁴, αφού κατάφερε να ξεγελάσει το 33% των κριτών (Schofield, 2014).

Ορισμός ΤΝ

Στο ερώτημα «*Τι είναι Τεχνητή Νοημοσύνη;*» οι ερευνητές του χώρου δίνουν πολλές διαφορετικές απαντήσεις, φαινόμενο που δεν απαντά σε άλλους επιστημονικούς χώρους, όπως η Φυσική, η Χημεία, η Ιατρική κ.ά. Ωστόσο, όλοι φαίνεται να συμφωνούν πως η ΤΝ είναι επιστήμη και όχι απλώς ένας κλάδος της τεχνολογίας λογισμικού.

Κατά τον Patrick Winston (1992), διευθυντής του εργαστηρίου ΤΝ του Πανεπιστημίου MIT, πρωταρχικός σκοπός της ΤΝ είναι «να κάνει τις μηχανές πιο έξυπνες» σε αυτό συμφωνούν οι περισσότεροι από τους ερευνητές που αντιμετωπίζουν την ΤΝ ως αναζήτηση μεθόδων οι οποίες θα κάνουν τους ηλεκτρονικούς υπολογιστές πιο έξυπνους και, συνεπώς, πιο χρήσιμους από όσο είναι σήμερα .

Όμως η ΤΝ έχει και πρακτικούς και επιστημονικούς στόχους.

Οι πρακτικοί στόχοι αποβλέπουν στο να επιλυθούν πραγματικά προβλήματα της ανθρώπινης επιβίωσης, με τη χρήση ιδεών της ΤΝ σχετικών με την αναπαράσταση και τη χρήση της απαραίτητης γνώσης. Ένας καλά διατυπωμένος ορισμός, που καλύπτει τους πρακτικούς στόχους της ΤΝ και, πιο συγκεκριμένα, του κλάδου της ΤΝ που καλείται Υπολογιστική Τεχνητή Νοημοσύνη, είναι ο ακόλουθος:

«Τεχνητή Νοημοσύνη είναι εκείνος ο κλάδος της επιστήμης των υπολογιστών που ασχολείται με το σχεδιασμό ευφυών υπολογιστικών συστημάτων, δηλαδή συστημάτων με χαρακτηριστικά τα οποία σχετίζονται με την ευφυΐα στην ανθρώπινη συμπεριφορά (μάθηση, αιτίαση, επίλυση προβλημάτων, κατανόηση φυσικής γλώσσας, αναγνώριση αντικειμένων κτλ.).»

³ <http://www.loebner.net/Prizef/loebner-prize.html>

⁴ <http://www.zdnet.com/article/computer-chatbot-eugene-goostman-passes-the-turing-test/>

Παρόμοιος είναι και ο ακόλουθος ορισμός:

«Τεχνητή Νοημοσύνη είναι η μελέτη τού πώς να κάνουμε τους υπολογιστές ικανούς να κάνουν πράγματα στα οποία προς το παρόν οι άνθρωποι τα καταφέρνουν καλύτερα» (Rich & Knight, 1990).

Οι επιστημονικοί στόχοι αποβλέπουν στο να καθοριστούν ποιες από τις υπάρχουσες ιδέες περί αναπαράστασης και χρήσης της γνώσης είναι σε θέση να δώσουν απαντήσεις στο αιώνιο φιλοσοφικό ερώτημα:

«τι είναι ευφυΐα και πώς αυτή εκφράζεται;»

Τα μοντέλα που χρησιμοποιεί η ΤΝ, με βάση οποιονδήποτε από τους δυο παραπάνω ορισμούς, στηρίζονται στη χρήση πολύπλοκων ηλεκτρονικών συστημάτων ως μέσου υλοποίησης. Το φαινόμενο αυτό εξηγείται βάσει της θεμελιώδους παράδοσης της δυτικής φιλοσοφίας ότι η νοητική ικανότητα (σκέψη) του ανθρώπου είναι στην ουσία ένας λογικός χειρισμός νοητικών συμβόλων, δηλαδή ιδεών. Ο ηλεκτρονικός υπολογιστής, σε αντίθεση με άλλα μηχανικά κατασκευάσματα (π.χ. ρολόι), μπορεί να χειριστεί σύμβολα σε μορφή “χαρακτήρων”, αφού πρώτα προγραμματιστεί κατάλληλα. Σύμφωνα με τη θεωρία χειρισμού συμβόλων, η νοημοσύνη εξαρτάται μόνο από την οργάνωση ενός συστήματος και τη λειτουργία του ως χειριστή συμβόλων και όχι από το υλικό κατασκευής των συμβόλων ή την ακριβή μορφή τους. Συνάγεται, λοιπόν, το συμπέρασμα πως η σύγχρονη τεχνολογία των υπολογιστών είναι προς το παρόν η αρμόζουσα, ώστε αυτοί να αποτελέσουν μοντέλα με δυνατότητες επίδειξης κάποιας μορφής τεχνητής νοημοσύνης. Ωστόσο, τίποτα δεν αποκλείει στο μέλλον κάποια άλλη τεχνολογία να αποδειχθεί καταλληλότερη για τον παραπάνω σκοπό.

Η παραπάνω θεώρηση οδηγεί στον ορισμό της λεγόμενης **συμβολικής Τεχνητής Νοημοσύνης** (symbolic Artificial Intelligence):

«Συμβολική Τεχνητή Νοημοσύνη είναι η επιστήμη που μελετά τη φύση της ανθρώπινης νοημοσύνης και στη συνέχεια τον τρόπο αναπαραγωγής της σε υπολογιστές με τη χρήση συμβόλων».

Το αντικείμενο της έρευνας στο χώρο της συμβολικής ΤΝ είναι η μελέτη των ανθρώπινων διαδικασιών σκέψης (νοημοσύνη, ευφυΐα, εμπειρία), στις οποίες θα αναφερθούμε εκτενέστερα παρακάτω, και των τρόπων αναπαράστασής τους μέσω μηχανών (ηλεκτρονικών υπολογιστών, ρομπότ κτλ.). Δεν αρκούν όμως μόνο οι μεθοδολογίες της επιστήμης των υπολογιστών. Καθοριστικά συμβάλλουν τόσο η ανάλυση όσο και διερεύνηση της ανθρώπινης ευφυούς συμπεριφοράς στην επίλυση προβλημάτων, τη χρήση και την κατανόηση της φυσικής γλώσσας και σε πολλούς άλλους χώρους, στους οποίους μπορούν να συνεισφέρουν σημαντική γνώση και άλλες επιστήμες, ιδιαίτερα η Ψυχολογία, η Γλωσσολογία και η Φιλοσοφία της Νόησης, οι οποίες σχετίζονται στενά με τις αφηρημένες αρχές της νοητικής οργάνωσης.

Το σύνολο των επιστημών που αναφέρονται στο σχήμα 1 αποτελούν πλέον τον ενοποιημένο γνωστικό κλάδο της Γνωστικής ή Γνωσιακής Επιστήμης (Cognitive Science), με πεδίο μελέτης τη γνωστική ικανότητα της ευφυΐας και της νόησης, δηλαδή το χειρισμό των συμβόλων. Βασικό εργαλείο της σχετικά νέας αυτής επιστήμης είναι προς το παρόν ο υπολογιστής, που θα μπορούσαμε ευφυολογώντας να παρομοιάσουμε με εργαστήρι, μέσα στο οποίο αναπτύσσονται οι σκέψεις σχετικά με τη σκέψη!



Σχήμα 1 Η σχέση της TN με τις άλλες σχετικές επιστήμες στον ευρύτερο χώρο της Γνωστικής Επιστήμης

Από την πλευρά της, η TN μπορεί να προσφέρει στο χώρο της Γνωστικής Επιστήμης μεθοδολογίες ανάπτυξης υπολογιστικών συστημάτων που υλοποιούν ή ακόμα δίνουν και λύσεις σε θεωρητικά μοντέλα απαραίτητα για την προώθηση της έρευνας και τη διαμόρφωση νέων θεωριών σχετικά με τη λειτουργία της ανθρώπινης νόησης. Δηλαδή, η TN ρίχνει νέο φως σε παραδοσιακά ερωτήματα των ψυχολόγων, γλωσσολόγων και φιλοσόφων.

Με βάση τα παραπάνω, ένας πληρέστερος ορισμός της TN είναι ο ακόλουθος:

«Τεχνητή Νοημοσύνη είναι η επιστήμη που επιδιώκει να κατασκευάσει μηχανές οι οποίες όχι μόνο θα επιδεικνύουν ανθρώπινη συμπεριφορά, αλλά θα μπορούν, επίσης, να προσαρμόζονται στο περιβάλλον τους με τρόπο παρόμοιο με αυτόν των ανθρώπων».

Υπολογιστική Νοημοσύνη

Ο όρος Τεχνητή Νοημοσύνη (TN) είναι γενικός, υπό την έννοια ότι υποδηλοί κάθε είδος μη βιολογικής νοημοσύνης. Ο συγγενής όρος **Υπολογιστική Νοημοσύνη-YN** (Computational Intelligence-CI), που αφορά τον κλάδο της TN με πρακτικούς στόχους, υποδηλοί κάθε είδος μη βιολογικής νοημοσύνης που μπορεί να εκδηλωθεί με υπολογιστικές διαδικασίες. Αποδεκτός ορισμός για την YN είναι ο ακόλουθος:

«Υπολογιστική Νοημοσύνη είναι ο επιστημονικός χώρος που προσφέρει τις τεχνικές για την επίλυση δύσκολων προβλημάτων, με τη μηχανή να μιμείται απλώς, βιολογικές διεργασίες, χωρίς να είναι απαραίτητο να επιδεικνύει γενική νοημοσύνη.».

Ο όρος χρησιμοποιήθηκε για πρώτη φορά από τον J. Bezdek (1992) στο επιστημονικό περιοδικό *International Journal of Approximate Reasoning*. Σε άρθρο του που αφορούσε τα τεχνητά νευρωνικά δίκτυα, ο Bezdek εισήγαγε τον όρο ABCs για να αποσαφηνίσει τα παρακάτω:

A = <i>Artificial</i>	Non - Biological (Man-Made)
B = <i>Biological</i>	Physical + Chemical + (??) = Organic
C = <i>Computational</i>	Mathematics + Man-Made Machines

Ο R. Marks (1993) ανέφερε για τη σχέση της YN και της TN:

«Παρόλο που επιδιώκουν παρόμοιους στόχους, η Υπολογιστική Νοημοσύνη αναδύθηκε ως ένας ανεξάρτητος κλάδος, του οποίου το ερευνητικό πεδίο είναι κάπως διαφορετικό από αυτό της Τεχνητής Νοημοσύνης».

Η ΥΝ χαρακτηρίζεται κυρίως από ορισμένες σημαντικές ιδιότητες, που είναι χαρακτηριστικά συστημάτων που χρησιμοποιούν τεχνικές μηχανικής μάθησης, όπως:

- προσαρμοστικότητα (adaptation),
- αυτο-οργάνωση (self-organization),
- μάθηση-εξέλιξη (learning-evolution).

Αρχικά η ΥΝ χρησιμοποιήθηκε σε εφαρμογές αναγνώρισης εικόνας (pattern recognition). Σήμερα χαρακτηρίζει εφαρμογές (ενδεικτικά):

- τεχνητών νευρωνικών δικτύων (artificial neural networks),
- εμπείρων συστημάτων (expert systems),
- τεχνητής ζωής (artificial life),
- γενετικών αλγορίθμων (genetic algorithms),
- ασαφούς λογικής (fuzzy logic).

Ιστορική Εξέλιξη της Τεχνητής Νοημοσύνης

Οι απαρχές της Τεχνητής Νοημοσύνης ανάγονται στους “συλλογισμούς” του Αριστοτέλη (384-322 π.Χ.), οι οποίοι παρείχαν πρότυπα εκφράσεων που έδιναν πάντα σωστά συμπεράσματα από σωστές υποθέσεις (Αριστοτέλεια συλλογιστική). Στους νεότερους χρόνους, οι πρώτες σημαντικές στιγμές είναι το 1854, οπότε ο George Boole έθεσε τις βάσεις της προτασιακής λογικής, και το 1879, οπότε ο Gottlob Frege πρότεινε ένα σύστημα αυτοματοποιημένης συλλογιστικής και έθεσε τις βάσεις του **κατηγορηματικού λογισμού** (predicate calculus).

Οι σημαντικές ημερομηνίες στην ιστορία της ΤΝ είναι:

1943-56 Η γέννηση της Τεχνητής Νοημοσύνης

1943 Οι McCulloch και Pitts προτείνουν ένα μοντέλο τεχνητών νευρώνων που έχει τη δυνατότητα να μαθαίνει και να υπολογίζει κάθε υπολογίσιμη συνάρτηση.

1950 Ο Alan Turing, που θεωρείται ο πατέρας της ΤΝ, εμπνέεται το τεστ της μίμησης (τεστ Τούρινγκ) για την αναγνώριση ευφών μηχανών.

1951 Οι Minsky και Edmonds υλοποιούν το πρώτο νευρωνικό δίκτυο, το SNARC (*Stochastic Neural Analog Reinforcement Calculator*), το οποίο έχει 40 νευρώνες και χρησιμοποιεί 3000 λυχνίες.

1956-70 Πρώτη Φάση ανάπτυξης της Τεχνητής Νοημοσύνης.

1956 Συνάντηση στο Dartmouth College ερευνητών από το χώρο των Μαθηματικών, της Ηλεκτρονικής και Ψυχολογίας (McCarthy, Allen Newell, Herbert Simon, Marvin Minsky) με κοινό στόχο τη μελέτη δυνατοτήτων χρήσης των υπολογιστών για την προσομοίωση της ανθρώπινης νοημοσύνης.

1958 Δημιουργία της γλώσσας Lisp από τον McCarthy.

1966 Μετά από έρευνα γύρω από την κατανόηση γλώσσας και την αντίληψη μηχανής, ο Weizenbaum δημιουργεί το ELIZA.

1970-80 Ωρίμαση της συμβολικής και υπολογιστικής Τεχνητής Νοημοσύνης.

1977 Δημιουργία των πρώτων εμπείρων συστημάτων: DENDRAL (1971), MYCIN (1975), Prospector (1977).

- 1972 α. Οι Colmerauer και Roussel από το Πανεπιστήμιο της Μασσαλίας σε συνεργασία με τον R. Kowalski από το Πανεπιστήμιο του Εδιμβούργου καταλήγουν στη δημιουργία της γλώσσας λογικού προγραμματισμού PROLOG.
β. Ο Winograd εμβαθύνει στην κατανόηση φυσικής γλώσσας.
- 1975 & 1977 Ο M. Minsky δημοσιεύει κεφάλαια περί αναπαράστασης της γνώσης σε βιβλία.
- 1976 Οι Newell & Simon υποστηρίζουν την υπόθεση ότι ένα φυσικό συμβολικό σύστημα διαθέτει τα απαραίτητα χαρακτηριστικά για νοήμονες ενέργειες.
- 1970 - Ανάπτυξη εξελικτικών αλγορίθμων.
Εκδίδονται βιβλία με μελέτες:
- 1973 του Rechenberg για τη βελτιστοποίηση των τεχνικών συστημάτων και τις αρχές της βιολογικής εξέλιξης.
- 1975 του Holland για την προσαρμοστικότητα στα φυσικά και τεχνητά συστήματα.
- 1992 του Koza, για το Γενετικό Προγραμματισμό (Genetic Programming).
- 1995 του Fogel για τον Εξελικτικό Υπολογισμό (Evolutionary Computation).
- 1980-90 Αναγέννηση των Τεχνητών Νευρωνικών Δικτύων.
- 1986 Οι Rumelhart and McClelland περιγράφουν τη δημιουργία προσομοιώσεων της αντίληψης στον υπολογιστή.
- 1987 1^ο Διεθνές Συνέδριο για τα Νευρωνικά Δίκτυα του IEEE (Institute of Electrical and Electronics Engineers).
- 1960 - : Αντιμετώπιση της ασάφειας στη γνώση.
- 1965 & 1968 Πρώτος ο Zadeh εισάγει τους όρους «Ασαφή Σύνολα» (Fuzzy Sets, 1965) και «Ασαφείς Αλγόριθμοι» (Fuzzy Algorithms, 1968).
- 1983 Ο Sugeno διατυπώνει την «Ασαφή Θεωρία».
- 1992 1^ο Συνέδριο του IEEE για τα Ασαφή Σύνολα.
- 1990 - Δημιουργία αφενός υπολογιστικών συστημάτων και μηχανών που βασίζονται σε αρχές της TN και τα οποία παρουσιάζουν τάσεις προσαρμογής στο περιβάλλον τους (π.χ. ρομπότ) και αφετέρου εφαρμογών που τείνουν να “μαθαίνουν” από την εμπειρία τους: Νοήμονες πράκτορες, Μηχανές Αναζήτησης στο διαδίκτυο, Περιρρέουσα Νοημοσύνη.

Η έρευνα στο χώρο της TN

Παρότι από την ίδρυση του κλάδου της TN έχουν περάσει περισσότερα από πενήντα χρόνια αδιάλειπτα έντονης ερευνητικής δραστηριότητας που έχει αποφέρει καρπούς, με αποτέλεσμα να έχει αλλάξει τεχνολογικά ο σύγχρονος κόσμος, ο χώρος εξακολουθεί να προσφέρει ερευνητικές προκλήσεις, με κεντρικό στόχο τη δημιουργία όλο και πιο ευφυών μηχανών. Το παρακάτω σχήμα παρουσιάζει τα τρία στρώματα στα οποία θα μπορούσε να χωριστεί ο χώρος της βασικής έρευνας που αφορά την TN.



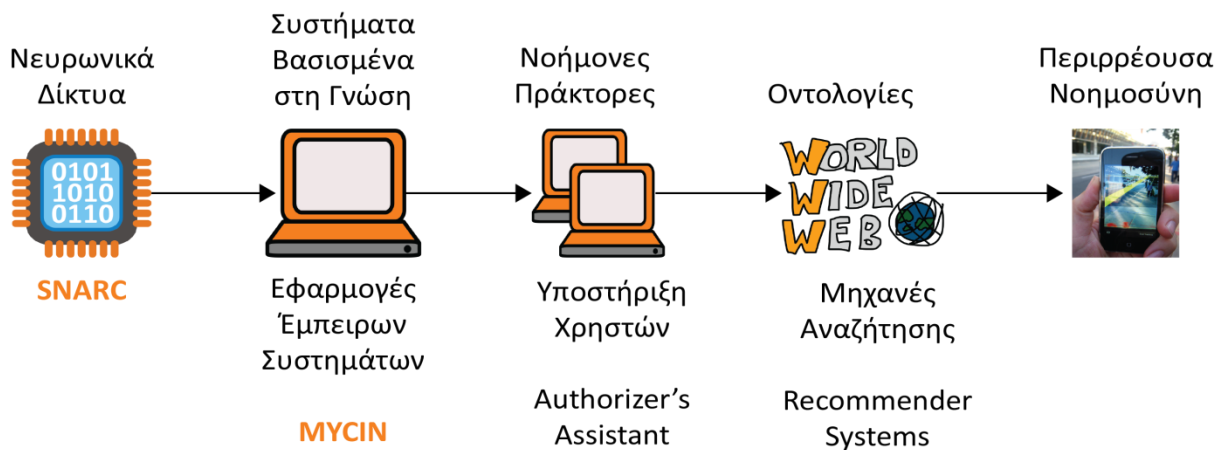
Σχήμα 2 Τα τρία στρώματα της έρευνας στο χώρο της TN

Στον πίνακα που ακολουθεί παρουσιάζονται οι σημαντικότεροι ερευνητικοί χώροι της Τεχνητής Νοημοσύνης σήμερα.

Αναπαράσταση Γνώσης (Knowledge Representation)	Μελετά το σύνολο των μοντέλων αναπαράστασης γνώσης στο χώρο της Γνωστικής Επιστήμης καθώς και τις μεθόδους επεξεργασίας τους.
Αντίληψη Μηχανής ή Μηχανική Όραση (Machine Vision)	Αφορά την αναγνώριση οπτικής εικόνας.
Επεξεργασία και Κατανόηση Φυσικής Γλώσσας (Natural Language Processing and Understanding)	Αφορά την επικοινωνία του χρήστη με τη μηχανή μέσω γραπτής αλλά και προφορικής φυσικής γλώσσας, καθώς και τη μετάφραση γλωσσών.
Μηχανισμοί Εξαγωγής Συμπερασμάτων – Έμπειρα Συστήματα (Expert Systems)	Εδώ εξετάζεται η ύπαρξη μηχανισμών που να χειρίζονται κατάλληλα τα γεγονότα και τους κανόνες, μέσω των οποίων αναπαρίσταται οι λογικοί συλλογισμοί, ώστε να εξάγονται σωστά συμπεράσματα.
Επίλυση Προβλημάτων (Problem Solving)	Στον τομέα αυτό μελετώνται ευφυείς αλγόριθμοι αναζήτησης λύσεων.
Μηχανική Μάθηση (Machine Learning)	Ασχολείται με το σχεδιασμό συστημάτων που μπορούν να μαθαίνουν μέσω επαγωγικών μεθόδων, όπως τα δένδρα απόφασης, να επιδεικνύουν δυνατότητες ελέγχου προτύπων, όπως τα νευρωνικά δίκτυα, και να αυτο-προσαρμόζονται βελτιστοποιώντας την απόδοσή τους, όπως οι γενετικοί αλγόριθμοι.
Σχεδιασμός Ενεργειών (Planning)	Εδώ μελετώνται τρόποι αποδοτικότερου σχεδιασμού ενεργειών και έχει εφαρμοστεί με επιτυχία σε προγραμματισμό παραγωγής σε βιομηχανίες και σε περιπτώσεις όπου πρέπει να αντιμετωπιστούν κρίσιμα γεγονότα.
Ρομποτική (Robotics)	Ασχολείται με την κίνηση, το χειρισμό και την αναγνώριση αντικειμένων από μηχανές.
Νοήμονες Πράκτορες (Intelligent Agents)	Προγράμματα που προσφέρουν ευφυή υποστήριξη σε άλλα προγράμματα ή στον χρήστη.
Ευφυείς Υπηρεσίες Διαδικτύου και Σημασιολογικού Ιστού (Semantic Web)	Ευφυείς διαδικτυακοί πράκτορες για υποστήριξη χρηστών στη χρήση του διαδικτύου και την πρόσβαση σε πληροφορίες του σημασιολογικού ιστού (π.χ. υπηρεσίες Big Data).
Προσαρμοζόμενα και Εξελισσόμενα Ευφυή Συστήματα	Μοντέρνα υβριδικά συστήματα μηχανικής μάθησης.
Ευφυή Εργαλεία (Intelligent Tools)	Εδώ περιλαμβάνονται οι γλώσσες προγραμματισμού TN και τα περιβάλλοντα ανάπτυξής τους.

Πίνακας 1 Ερευνητικοί χώροι στο χώρο της TN

Με την εξάπλωση του διαδικτύου και τον μεγάλο όγκο πληροφοριών αποθηκευμένων εκεί, έχει αναπτυχθεί πληθώρα συστημάτων βασισμένων σε τεχνικές της TN και, ιδιαίτερα, υβριδικών συστημάτων μηχανικής μάθησης, που συνδυάζουν έμπειρα συστήματα με ασαφή λογική, νευρωνικά δίκτυα και γενετικούς αλγορίθμους. Γνωστά τέτοια συστήματα είναι οι **νοήμονες πράκτορες** (intelligent agents), οι **μηχανές ευφυούς αναζήτησης** (intelligent search machines), τα **συστήματα εξόρυξης δεδομένων** (data mining systems), τα **συμβουλευτικά συστήματα** (recommender systems) και τα **συστήματα περιρρέουσας νοημοσύνης** (ambient intelligence systems).



Σχήμα 3 Βασικά στάδια εξέλιξης της TN

Για να είναι αποτελεσματικά τα παραπάνω συστήματα, νέες μέθοδοι αναπαράστασης γνώσης έχουν καθιερωθεί, όπως είναι τα σημασιολογικά δίκτυα και οι οντολογίες.

Τεχνητή και Περιρρέουσα Νοημοσύνη

Το όραμα της εξέλιξης της κοινωνίας της γνώσης, όπου δίνεται έμφαση στη μεγαλύτερη φιλικότητα των υπολογιστικών συστημάτων προς τον χρήστη, την υποστήριξη πιο αποτελεσματικών υπηρεσιών, την ενδυνάμωση του χρήστη και την υποστήριξη αλληλεπιδράσεων μεταξύ μηχανών και ανθρώπων, φαίνεται να γίνεται πραγματικότητα μέσω των εφαρμογών της **Περιρρέουσας Νοημοσύνης** (Ambient Intelligence-Aml): οι άνθρωποι περιβάλλονται από **ευφυείς διεπαφές χρήσης** (intelligent intuitive interfaces), που είναι ενσωματωμένες σε όλα τα καθημερινά αντικείμενα.

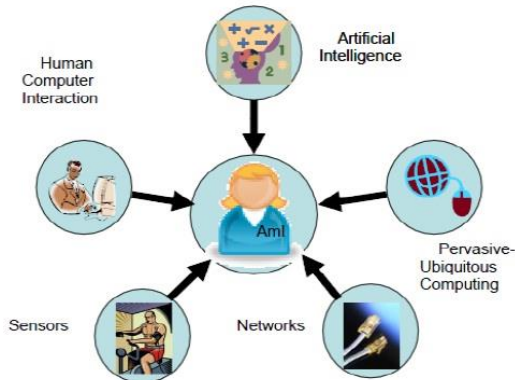


Σχήμα 4 Αλληλεπίδραση με αντικείμενα σε περιβάλλον Περιρρέουσας Νοημοσύνης⁵

Το περιβάλλον Περιρρέουσας Νοημοσύνης είναι σε θέση να αναγνωρίζει την παρουσία κάθε ανθρώπου και να προσαρμόζεται σε αυτήν, αλλά οι άνθρωποι δεν συνειδητοποιούν πως βρίσκονται σε υπολογιστικό περιβάλλον· η Περιρρέουσα Νοημοσύνη είναι συνήθως «αόρατη». Δηλαδή, η έμφαση δίνεται στην ευκολία χρήσης και στην υποστήριξη και ενδυνάμωση της αλληλεπίδρασης των ανθρώπων με το περιβάλλον (Ducatel κ.ά., 2001). Η Περιρρέουσα Νοημοσύνη έχει δύο επίπεδα, το λειτουργικό και το νοήμον:

⁵ Από κείμενα της πρωτοβουλίας [Disappearing Computer](#) και του έργου e-gadgets

- Λειτουργικό Επίπεδο : Υλικά, λειτουργικά συστήματα, δίκτυα επικοινωνίας, βάσεις δεδομένων, γραφικά Η/Υ, διάχυτη πανταχού παρούσα υπολογιστική (pervasive-ubiquitous computing).
- Νοήμιον Επίπεδο: Αναπαράσταση γνώσης, λογική, οντολογίες, αναγνώριση ομιλίας, επεξεργασία φυσικής γλώσσας, έμπειρα συστήματα, μάθηση μηχανής, υπολογιστική νοημοσύνη κ.ά.



Σχήμα 5 Το περιβάλλον της AmI⁶

Σύγχρονες εφαρμογές της ΤΝ

Οι εφαρμογές που παρουσιάζουν το μεγαλύτερο ενδιαφέρον στη σύγχρονη Τεχνητή Νοημοσύνη είναι αυτές που ενδιαφέρουν το χώρο της Γνωστικής Επιστήμης και έχουν σχέση με τη μάθηση, οι εφαρμογές ρομποτικής και οι εφαρμογές φυσικών διεπαφών.



Σχήμα 6 Τομείς σύγχρονων εφαρμογών ΤΝ

Στις εφαρμογές Γνωστικής Επιστήμης συγκαταλέγονται εφαρμογές εμπειρών συστημάτων, προσαρμοστικά ευφυή περιβάλλοντα μάθησης, συστήματα που διαχειρίζονται ασαφή λογική, εφαρμογές που υποστηρίζονται από συστήματα μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και οι γενετικοί αλγόριθμοι, καθώς και ευφυείς πράκτορες.

Η κατηγορία των ρομποτικών εφαρμογών αφορά μηχανές που έχουν ικανότητες οπτικής αντίληψης, αφής, επιδεξιότητας, μετακίνησης μελών και πλοήγησης στο χώρο.

⁶ Από www.seminarsonly.com

Τέλος, στην τρίτη κατηγορία ανήκουν οι εφαρμογές που μπορούν να επικοινωνούν με φυσική γλώσσα, με αναγνώριση φωνής και τα περιβάλλοντα διεπαφής τους μπορούν να διαθέτουν πολλαπλούς αισθητήρες δημιουργώντας την αίσθηση περιρρέουσας νοημοσύνης. Στην κατηγορία αυτή ανήκουν όλα τα περιβάλλοντα εικονικής πραγματικότητας.

Βιβλιογραφία/Αναφορές

- Bezdek, J. C. (1992). On the Relationship between Neural Networks, Pattern Recognition and Intelligence. *Int. J. Approximate Reasoning*, 6(2), 85-107.
- Boole, G. (1854). *An Investigation of the Laws of Thought on which are founded the mathematical theories of logic and probabilities*. Λονδίνο: Walton & Maberly. Ανακτήθηκε από <http://history-computer.com/Library/Boole.pdf>
- Colmerauer, A. , & Roussel, P. (1996). The birth of Prolog. Στο T.J.Bergin & R.G.Gibson (επιμ.) *History of programming languages-II* (σ.331-367). Νέα Υόρκη: ACM Press.
- Ducatel, K., Bogdanowicz, M.,Scapolo, F.,Leijten, J. & Burgelman, J-C. (2001) . *Scenarios for Ambient Intelligence in 2010*. Ανακτήθηκε από <ftp://ftp.cordis.europa.eu/pub/ist/docs/istagscenarios2010.pdf>.
- Fogel, D. B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, Νιου Τζέρσεϊ: IEEE Press.
- Frege,G. (1879). *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Hallea. S.: Louis Nebert.
- Hodges, A. (1995). *Alan Turing: a short biography*. Ανακτήθηκε από <http://www.turing.org.uk/publications/dnb.html>
- Holland, J.H. (1975/1992). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press /Καίμπριτζ, Μασσαχουσέττη :MIT Press.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- Koza, J. R. (1992a). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Καίμπριτζ, Μασσαχουσέττη: MIT Press.
- Marks, R. (1993). Intelligence: Computational versus Artificial. *IEEE Transactions on Neural Networks*, 4(5), 737-739. McCarthy, J. (1960). Recursive Functions of Symbolic Expressions and Their Computation by Machine (Part I). *Communications of the ACM*, 3(4),184-195. Ανακτήθηκε από <http://www-formal.stanford.edu/jmc/recursive.pdf>
- McCulloch, W.S., & Pitts,W.H.(1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133. Ανακτήθηκε από <http://www.minicomplexity.org/pubs/1943-mcculloch-pitts-bmb.pdf>
- Minsky, M. (1975). A Framework for Representing Knowledge. Στο P.H.Winston (εκδ.), *The Psychology of Computer Vision* . Νέα Υόρκη: McGraw-Hill.
- Minsky, M. (1977). Frame theory. Στο P.N. Johnson-Laird & P.C.Wason (επιμ.), *Thinking: Reasings in Cognitive Science* (σ 355-376). Καίμπριτζ: Cambridge University Press.
- Newell, A., & Simon, H.A.(1976). Computer science as empirical inquiry: symbols and search. *Communications of the ACM*,19 (3), 113-126.

- Rechenberg, I.(1973). *Evolutionsstrategie: Optimierung technischer Systeme und Prinzipien der biologischen Evolution*. Στουτγκάρδη: Frommann-Holzboog.
- Rich, E., & Knight, K. (1990). *Artificial Intelligence* (2^η έκδοση). McGraw-Hill Higher Education.
- Rich, E. Knight, K. & Nair, S.B. (2009). *Artificial Intelligence* (3^η έκδοση). Νέο Δελχί: Tata McGraw-Hill Higher Education.
- Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (2 τόμοι). Καίμπριτζ, Μασσαχουσέτη: MIT Press.
- Schofield, J. (2014). Computer chatbot 'Eugene Goostman' passes the Turing test. Ανακτήθηκε από <http://www.zdnet.com/article/computer-chatbot-eugene-goostman-passes-the-turing-test/>
- Sugeno, M. (1983). *Fuzzy theory IV* (σημειώσεις διάλεξης στα ιαπωνικά), *J. SICE* 22, 554-559.
- Turing, A. (1950). Computing Machinery and Intelligence, *Mind*, 59, 433-460. Ανακτήθηκε από <http://www.loebner.net/Prizef/TuringArticle.html>
- Winston, P. H. (1992). *Artificial Intelligence* (3^η έκδοση). Reading, Μασσαχουσέτη : Addison-Wesley. Ανακτήθηκε από <http://courses.csail.mit.edu/6.034f/ai3/rest.pdf>
- Zadeh, L.A. (1965). Fuzzy sets, *Information and Control*, 8 (3), 338–353.
- Zadeh, L.A. (1968). Fuzzy Algorithms, *Information and Control*, 12, 99-102.

ΚΕΦΑΛΑΙΟ 1 - Επίλυση Προβλημάτων

Σύνοψη

Στο κεφάλαιο αυτό θα παρουσιαστούν με παραδειγματικές περιπτώσεις οι θεμελιώδεις έννοιες για τον ορισμό ενός προβλήματος και η επίλυσή του μέσω αλγορίθμων αναζήτησης, κατά την οποία μπορεί ένα πρόγραμμα να βρει με εναλλακτικούς τρόπους μια ακολουθία ενεργειών που επιτυγχάνει τους στόχους του προβλήματος. Στη συνέχεια, θα περιγραφούν οι εναλλακτικοί τρόποι αναζήτησης με εμβάθυνση στην αναζήτηση με χώρο καταστάσεων. Τέλος, θα παρουσιαστεί η γενική αλγοριθμική προσέγγιση για τους αλγόριθμους αναζήτησης, θα παρουσιαστούν αναλυτικά οι πιο γνωστοί από αυτούς και θα συγκριθούν τα πλεονεκτήματά και τα μειονεκτήματά τους.

Προαπαιτούμενη γνώση

Δεν υπάρχει προαπαιτούμενη γνώση.

1.1 Προβλήματα και αναζήτηση λύσης

Όταν επιχειρούμε να λύσουμε ένα πρόβλημα, πρέπει πρώτα να το ορίσουμε και ακολούθως να σχεδιάσουμε έναν αλγόριθμο ο οποίος θα το επιλύει έχοντας στη διάθεσή του πολλές άμεσες επιλογές ενεργειών σε κάθε βήμα του και αποφασίζοντας ποια να επιλέξει ή στα τυφλά ή βάσει ενός ποιοτικού κριτηρίου.

Αναζήτηση (search) ονομάζεται η διαδικασία εύρεσης μιας ακολουθίας επιλεγμένων ενεργειών που οδηγεί σε ένα στόχο. Ένας **αλγόριθμος αναζήτησης** (search algorithm) δέχεται ως είσοδο ένα πρόβλημα και επιστρέφει μια λύση με τη μορφή ακολουθίας ενεργειών. Για να δοθεί ένα πρόβλημα ως είσοδος σε έναν αλγόριθμο αναζήτησης, πρέπει να έχει προηγηθεί μια σαφής διατύπωσή του και βάσει αυτής να γίνει ο ορισμός του με τρόπο κατάλληλο, ώστε να αποτελέσει είσοδο στον αλγόριθμο.

Τα συστήματα της ΤΝ αναζητούν λύσεις μέσα από εναλλακτικές αλγοριθμικές προσεγγίσεις και το κεφάλαιο που ακολουθεί αναφέρεται σε κάποιες από τις ευρέως χρησιμοποιούμενες για το σκοπό αυτόν. Πριν, όμως, αναφερθούμε σε συγκεκριμένους αλγόριθμους, θα πρέπει να μελετήσουμε τους διαφορετικούς τύπους των προς επίλυση προβλημάτων, δεδομένου ότι από αυτούς εξαρτάται άμεσα η καταλληλότητα ενός αλγόριθμου.

1.1.1 Τύποι προβλημάτων

Σε ορισμένα προβλήματα, ο στόχος προς επίτευξη είναι ήδη γνωστός και τα επιμέρους βήματα που θα οδηγήσουν σε αυτόν είτε δεν μας απασχολούν (παράδειγμα: στο τάβλι ο στόχος είναι να κερδίσει ένας από τους δύο παίκτες χωρίς να μας ενδιαφέρει να αποτυπωθεί το πώς το πέτυχε) είτε αποτελούν καίριο ζητούμενο (παράδειγμα: στο πρόβλημα του περιοδεύοντος πωλητή δε μας αρκεί απλώς να καταφέρει να φτάσει ο πωλητής στην πόλη που αποτελεί το στόχο του προβλήματος αλλά μας ενδιαφέρει μετακινούμενος από πόλη σε πόλη μέχρι να φτάσει στη πόλη-στόχο να διανύσει την μικρότερη δυνατή απόσταση). Σε άλλη κατηγορία προβλημάτων, ο στόχος δεν είναι γνωστός και μας ενδιαφέρει το πού μπορεί να καταλήξει η αναζήτηση (σχετικό παράδειγμα είναι το πρόβλημα της διάγνωσης ασθένειας βάσει των συμπτωμάτων του ασθενούς).

Οι αλγόριθμοι με συγκεκριμένη στόχευση έχουν μεγαλύτερες δυνατότητες μεγιστοποίησης της απόδοσής τους από αλγόριθμους χωρίς στόχευση. Ο στόχος ενός προβλήματος μπορεί να αντιστοιχεί είτε σε μία μόνη τελική κατάστασή του (για παράδειγμα, στο πρόβλημα του περιοδεύοντος πωλητή η τελική κατάσταση είναι μια συγκεκριμένη πόλη) είτε σε εναλλακτικές τελικές καταστάσεις του προβλήματος (στο τάβλι ο στόχος είναι το να πάρει όλα του τα πούλια στο χέρι ο ένας από τους δύο παίκτες, αλλά, όταν ο στόχος επιτευχθεί, τα πόνια του άλλου παίκτη που έχουν απομείνει μπορεί να βρίσκονται σε πολλές εναλλακτικές θέσεις, κάθε πιθανός συνδυασμός των οποίων αποτελεί και μια δεκτή κατάσταση ως τελική).

Οι διαφορετικοί τύποι προβλημάτων είναι εν γένει οι ακόλουθοι:

1. Προβλήματα στα οποία είναι πλήρως γνωστά ο στόχος και οι αντίστοιχες εναλλακτικές τελικές καταστάσεις τους και επιδιώκεται να βρει ο αλγόριθμος μία από αυτές: π.χ. προβλήματα σχεδιασμού ενεργειών και προβλήματα πλοήγησης.
2. Προβλήματα στα οποία είναι γνωστά ο στόχος και οι αντίστοιχες εναλλακτικές τελικές καταστάσεις τους και επιδιώκεται να βρει ο αλγόριθμος τη βέλτιστη από αυτές: π.χ. προβλήματα βελτιστοποίησης.
3. Προβλήματα στα οποία είναι γνωστές μόνο κάποιες ιδιότητες των τελικών καταστάσεων και επιδιώκεται η εύρεση ενός πλήρους στιγμιότυπου τελικής κατάστασης: π.χ. προβλήματα χρονοπρογραμματισμού, γνωστά ως προβλήματα ικανοποίησης περιορισμών.
4. Προβλήματα στα οποία ο στόχος δεν είναι γνωστός και επιδιώκεται να βρεθεί μια έγκυρη τελική κατάσταση: π.χ. προβλήματα εξαγωγής συμπερασμάτων.

Στο κεφάλαιο αυτό θα εστιάσουμε μόνο στον τύπο προβλημάτων που ανήκουν στην πρώτη κατηγορία. Ως **στόχος προβλήματος** (problem goal) μπορεί να οριστεί το σύνολο των καταστάσεων του προβλήματος στις οποίες το ζητούμενο εκπληρώνεται και οι οποίες χαρακτηρίζονται ως τελικές καταστάσεις του προβλήματος. Η διατύπωση του στόχου είναι το πρώτο βήμα για την επίλυση ενός προβλήματος αυτής της κατηγορίας.

Συνήθως, προβλήματα του παραπάνω τύπου μπορούν να επιλυθούν με οποιαδήποτε αλγοριθμική προσέγγιση· παράδειγμα, το πρόβλημα των *Πύργων του Ανόι* (βλέπε [Σχήμα 1.8](#)) επιλύεται εύκολα με αναδρομή, χωρίς να μας απασχολούν τα βήματα επίλυσης που οδηγούν στο στόχο. Το μειονέκτημα της προσέγγισης που βασίζεται σε αναδρομή παρόμοιων αλγοριθμικών προσεγγίσεων είναι ότι αφενός δεν αποτυπώνει ρητά την ακολουθία των ενεργειών που κάνει ο αλγόριθμος και αφετέρου δεν έχει δυνατότητες αναζήτησης βάσει ποιοτικών κριτηρίων. Η αντίστοιχη αλγοριθμική προσέγγιση στο χώρο της TN είναι οι αλγόριθμοι αναζήτησης που, όπως προαναφέραμε, αποτυπώνουν ρητά την ακολουθία των ενεργειών που εκτελούνται κατά την επίλυση.

1.1.2 Προβλήματα που επιλύονται με αναζήτηση

Στην περίπτωση ενός προβλήματος που επιλύεται με αναζήτηση, ο αλγόριθμος αναζήτησης βασίζεται πάντα στην κατάσταση στην οποία βρίσκεται το πρόβλημα: στο πρόβλημα του περιοδεύοντος πωλητή όταν γίνεται μια συγκεκριμένη ενέργεια, το αποτέλεσμα μιας ενέργειας μετακίνησης εξαρτάται άμεσα από την τρέχουσα κατάσταση του προβλήματος, δηλαδή από το σε ποια πόλη βρίσκεται ο πωλητής.

Πρωταρχικής σημασίας είναι η έννοια της **κατάστασης ενός προβλήματος** (problem state) και η έννοια του **χώρου καταστάσεων** (state space) ο οποίος είναι το σύνολο των έγκυρων καταστάσεων στις οποίες μπορεί να βρεθεί ένα πρόβλημα, συμπεριλαμβανομένων της αρχικής και των πιθανών τελικών καταστάσεων του. Τα προβλήματα που επιλύονται με αναζήτηση χαρακτηρίζονται και ως προβλήματα που επιλύονται με χώρο καταστάσεων.

Σε πολλά από τα προβλήματα που επιλύονται με αναζήτηση μας ενδιαφέρει η εύρεση της καλύτερης ποιοτικά λύσης, δηλαδή η αναζήτηση διαμορφώνεται ως πρόβλημα βελτιστοποίησης· στο πρόβλημα του περιοδεύοντος πωλητή απαιτείται να βρεθεί ο συντομότερος δρόμος που οδηγεί στο στόχο του προβλήματος. Η έννοια της καλύτερης λύσης εξαρτάται από το πρόβλημα. Πιθανόν να υponοείται η μεγιστοποίηση μιας μέτρησης (π.χ. κέρδος) ή η ελαχιστοποίηση κάποιου μεγέθους (π.χ. κόστος). Πολλές φορές, η έννοια του καλύτερου είναι σύνθετη και προκύπτει θετικά από την ποιότητα της τελικής κατάστασης που βρέθηκε και αρνητικά από το κόστος των ενεργειών που έγιναν (κόστος διαδρομής).

$$\text{Συνολικό κόστος} = \text{κόστος (διαδρομής)} - \text{κέρδος(κατάστασης στόχου)}$$

Τα προβλήματα που θα ερευνήσουμε ακολούθως ως υποδείγματα θα είναι προσδιοριστικά, δηλαδή οι ενέργειες που καλείται ο αλγόριθμος να κάνει σε κάθε βήμα του προσδιορίζονται επακριβώς από τους εγγενείς περιορισμούς του προβλήματος και όχι από εξωτερικούς παράγοντες. Υπάρχουν ωστόσο και προβλήματα στο πεδίο εφαρμογών της TN που σχετίζονται με παίξιμο παιχνιδιών (game playing) και στα οποία ο εξωτερικός κόσμος, μέσω των επιλογών του, επιδρά στη διαμόρφωση της τρέχουσας κατάστασης του προβλήματος· παράδειγμα είναι τα παιχνίδια δύο παικτών (σκάκι, τάβλι, τρίλιζα κτλ.) όπου ο 1ος παίκτης

είναι ο υπολογιστής, αλλά ο 2ος παίκτης είναι φυσικό πρόσωπο. Η παρουσία ενός παίκτη-αντιπάλου αλλάζει δραστικά την αναζήτηση, καθώς τώρα υπάρχει ο ρόλος του ενός παίκτη (του υπολογιστή) υπέρ του οποίου ο αλγόριθμος προσπαθεί να βρει την καλύτερη λύση, δηλαδή να κερδίσει, ενώ ο αντίπαλος προσπαθεί να τον σταματήσει. Στο τάβλι, επιπλέον εξωτερικός παράγοντας είναι η τύχη, η οποία μέσω του τυχαίου αποτελέσματος του ριξίματος των ζαριών διαμορφώνει με μη προσδιοριστικό τρόπο την επόμενη κατάσταση στην οποία θα βρεθεί το πρόβλημα. Όμως, σε κάθε περίπτωση τον αλγόριθμο αναζήτησης απασχολεί η τρέχουσα κατάσταση του προβλήματος. Σε ορισμένα άλλα προβλήματα, για παράδειγμα σε παιχνίδια τράπουλας όπως το μπριτζ, ο παίκτης πρέπει, επιπλέον, να «θυμάται» όλες τις προηγούμενες κινήσεις, είτε δικές του είτε των αντιπάλων του, ώστε να επιλέξει την επόμενη κίνησή του.

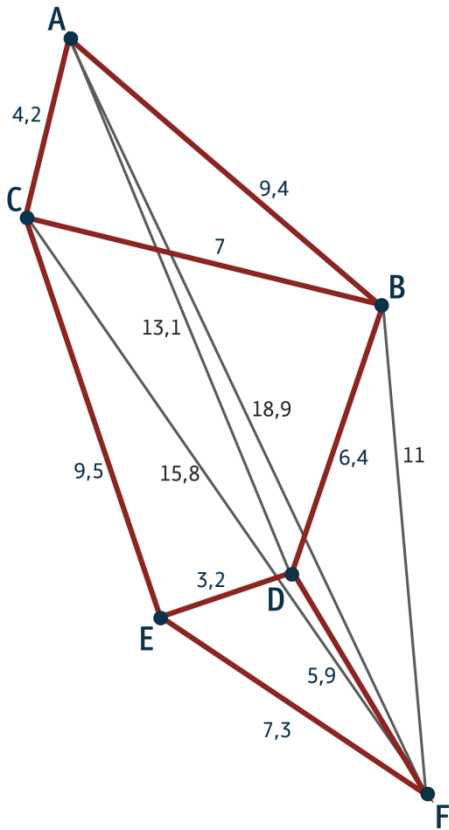
Η αλληλεπίδραση με το φυσικό περιβάλλον μπορεί να θεωρηθεί, επίσης, ως μια μορφή παιχνιδιού όπου οι εξωγενείς παράγοντες πολλές φορές δεν είναι καν γνωστοί εκ των προτέρων. Καθώς ο αλγόριθμος προσπαθεί να ενεργήσει στον πραγματικό κόσμο, ανακαλύπτει νέα γνώση και προκύπτουν συνθήκες που μπορούν να υποστηρίξουν ή να εμποδίσουν την προγραμματισμένη διαδικασία.

Το παρόν κεφάλαιο θα αφιερωθεί μόνο στην προσδιοριστική αναζήτηση λύσεων σε χώρο καταστάσεων σε μικρά υποδειγματικά προβλήματα, όπως τα προβλήματα περιορισμένου φυσικού κόσμου, παιχνιδιών ενός παίκτη και τα κουίζ. Απαραίτητο επόμενο βήμα, για να μπορέσουν να επιλυθούν παρόμοια προβλήματα, είναι να έχουν γίνει κατανοητά, δηλαδή να διατυπωθούν με σαφή τρόπο. **Διατύπωση ενός προβλήματος** (problem formulation) είναι η διαδικασία με την οποία αποφασίζεται ποιες ενέργειες και καταστάσεις θα πρέπει να εξετάζονται με δεδομένο ένα στόχο.

Στο χώρο της ΤΝ, η αναζήτηση απαιτεί η διατύπωση του προβλήματος να αποτυπώνεται σε έναν σαφή και τυποποιημένο ορισμό.

1.1.3 Ορισμός Προβλήματος

Για να οριστεί ένα πρόβλημα, πρέπει πρώτα να περιγραφεί ο κόσμος του προβλήματος. Ο **κόσμος του προβλήματος** (problem world) είναι ένα υποσύνολο του πραγματικού κόσμου και περιέχει μόνο τα αντικείμενα που έχουν άμεση σχέση με το πρόβλημα, τις ιδιότητές τους και τις σχέσεις που τα συνδέουν. Για παράδειγμα, στον κόσμο του προβλήματος του περιοδεύοντος πωλητή σε 6 πόλεις, που φαίνεται στο παρακάτω σχήμα 1.1, περιέχονται οι 6 πόλεις που μπορεί να επισκεφτεί ο πωλητής, οι ιδιότητες που έχουν, όσον αφορά την απόσταση κάθε πόλης από καθεμία από τις υπόλοιπες πόλεις και τις σχέσεις που τις συνδέουν, δηλαδή τους υπάρχοντες δρόμους, που σημειώνονται με κόκκινη γραμμή.



Σχήμα 1.1 Γραφική αναπαράσταση του κόσμου του προβλήματος του περιοδεύοντος πωλητή σε 6 πόλεις

Ο κόσμος ενός προβλήματος χαρακτηρίζεται ως:

- **κλειστός** (closed world), όταν κανένα νέο αντικείμενο, ιδιότητα ή σχέση δεν εισάγεται ή δεν εξάγεται από ή προς τον εξωτερικό κόσμο,
- **ανοιχτός** (open world), όταν δέχεται επιδράσεις από τον εξωτερικό κόσμο που επιδρούν στην τρέχουσα κατάστασή του· αυτό είναι πολύ δύσκολο να το διαχειριστούμε, γιατί καθιστά την εξέλιξη του κόσμου μη προβλέψιμη, όπως αναφέραμε και στην προηγούμενη παράγραφο.

Τα 4 χαρακτηριστικά στοιχεία που προκύπτουν από τη διατύπωση ενός προβλήματος και μπορούν να το ορίσουν τυπικά είναι:

- η **αρχική κατάσταση** (start state ή input state), από την οποία θα αρχίσει η αναζήτηση.
- ο **στόχος** (goal), ο οποίος περιλαμβάνει έγκυρες καταστάσεις με συγκεκριμένα επιθυμητά χαρακτηριστικά τις οποίες θέλει να εντοπίσει η αναζήτηση. Μια τέτοια κατάσταση χαρακτηρίζεται ως **κατάσταση-στόχος** (goal state) του προβλήματος.
- η **περιγραφή των ενεργειών** που έχει στη διάθεσή του ο αλγόριθμος. Αν δώσουμε το όνομα **συνάρτηση διαδόχων** (successor function) στη συνάρτηση που καλεί ένα σύνολο ενεργειών οι οποίες εφαρμόζονται σε μια κατάσταση, για να παράγουν διάδοχες καταστάσεις, τότε με δεδομένη μια συγκεκριμένη κατάσταση x του προβλήματος, η συνάρτηση διαδόχων θα επιστρέφει ένα σύνολο διατεταγμένων ζευγών (ενέργεια, διάδοχος), όπου κάθε ενέργεια είναι μία από τις επιτρεπτές ενέργειες στην κατάσταση x και διάδοχος (successor) είναι μια κατάσταση στην οποία μπορούμε να φτάσουμε από την κατάσταση x με την εκτέλεση αυτής της ενέργειας. Οι ενέργειες καλούνται και τελεστές μετάβασης (transition operators) του προβλήματος.
- ο **χώρος καταστάσεων** (state space) που ορίζεται έμμεσα από την αρχική κατάσταση και τη συνάρτηση διαδόχων. Ο χώρος καταστάσεων απεικονίζεται με ένα γράφημα του οποίου οι

κόμβοι είναι καταστάσεις και τα τόξα μεταξύ καταστάσεων είναι ενέργειες (βλέπε [Σχήμα 1.3](#)). Σε προβλήματα διάσχισης δρόμου το γράφημα του χώρου καταστάσεων ταυτίζεται με τη γραφική αναπαράσταση του κόσμου του προβλήματος (βλέπε [Σχήμα 1.1](#)).

Με βάση τα παραπάνω, μπορούμε να πούμε ότι ένα πρόβλημα (P) ορίζεται ως $P=(I,G,T,S)$, όπου:

I: η αρχική κατάσταση,

G: το σύνολο των καταστάσεων που περιλαμβάνει ο στόχος,

T: το σύνολο των τελεστών μετάβασης,

S: ο χώρος των καταστάσεων.

Στα τα προβλήματα όπου μας ενδιαφέρει η ποιότητα της λύσης, επιπλέον στοιχείο αποτελεί η **συνάρτηση κόστους διαδρομής** (function of path cost), όπως αυτή περιγράφηκε στην προηγούμενη [παράγραφο 1.1.2](#).

Στη συνέχεια, θα παρουσιάσουμε τη λεκτική περιγραφή των κλειστών κόσμων υποδειγματικών προβλημάτων, όπως του προβλήματος των 3 κύβων, του αγρότη με τη βάρκα, των 8 πλακιδίων, των 2 δοχείων κ.ά. Οι τεχνικές αναπαράστασης της γνώσης για ένα πρόβλημα, όπως αυτή προκύπτει μέσω της λεκτικής περιγραφής του, θα εκτεθούν για μερικά από τα παραπάνω προβλήματα στο κεφάλαιο της Αναπαράστασης της Γνώσης ([Κεφάλαιο 2](#)).

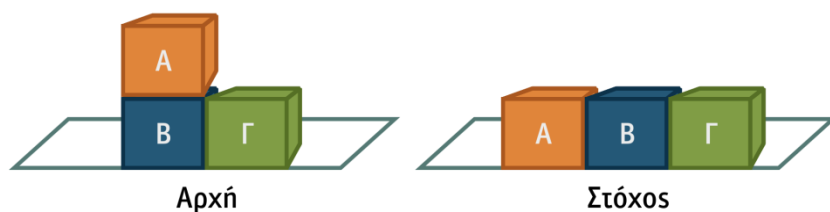
1.1.4 Παραδείγματα προβλημάτων

Εδώ θα παρουσιάσουμε μερικά από τα γνωστότερα προβλήματα που χρησιμοποιούνται ως παραδείγματα σε εκπαιδευτικά εγχειρίδια, κάνοντας διάκριση μεταξύ των προβλημάτων-παιχνιδιών και των προβλημάτων του πραγματικού κόσμου.

Προβλήματα-παιχνίδια

Το πρώτο παράδειγμα που θα εξετάσουμε είναι αυτό του προβλήματος των **3 κύβων**.

Διατύπωση του προβλήματος των 3 κύβων: Τρεις κύβοι βρίσκονται σε μια τυχαία διάταξη πάνω σε ένα τραπέζι. Σκοπός του προβλήματος είναι να μετακινηθούν οι κύβοι, ώστε να σχηματιστεί μια νέα επιθυμητή διάταξη. Ένας κύβος μπορεί να μετακινηθεί, εφόσον έχει ελεύθερη την επάνω έδρα του.



Σχήμα 1.2 Γραφική περιγραφή του κόσμου του προβλήματος των 3 κύβων

Καταστάσεις: Οι κύβοι βρίσκονται σε οποιαδήποτε δυνατή διάταξη. Επομένως, υπάρχουν 13 έγκυρες καταστάσεις (6 συνδυασμοί κύβων σε διάταξη στήλης, 6 συνδυασμοί κύβων σε διάταξη 2 κύβων σε στήλη και ενός μόνου του και 1 συνδυασμός των τριών κύβων σε διάταξη όλων πάνω στο τραπέζι).

Αρχική κατάσταση: Οποιαδήποτε κατάσταση μπορεί να οριστεί ως αρχική.

Τελεστές μετάβασης: Δύο ενέργειες επιτρέπεται να δοκιμαστούν σε κάθε κατάσταση: η μετακίνηση ενός ελεύθερου κύβου πάνω σε έναν άλλον ελεύθερο κύβο (move-to-cube) και η

μετακίνηση ενός ελεύθερου κύβου που βρίσκεται στην κορυφή μιας στήλης πάνω στο τραπέζι (move-to-table). Ο πλήρης χώρος καταστάσεων παρουσιάζεται στο [σχήμα 1.3](#).
Στόχος: Μια οποιαδήποτε διάταξη των κύβων μπορεί να τεθεί ως στόχος.

Το πρόβλημα των 3 κύβων, σε ένα στιγμιότυπό του, θα μπορούσε να περιγραφεί με τη βοήθεια του παρακάτω πίνακα:

Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος Α	Κύβος Α ελεύθερος	Κύβος Α πάνω σε κύβο Β
Κύβος Β	Κύβος Β όχι ελεύθερος	Κύβος Β πάνω σε Τ
Κύβος Γ	Κύβος Γ ελεύθερος	Κύβος Γ πάνω σε Τ
Τραπέζι Τ	Το τραπέζι διαθέτει επαρκή χώρο	

Πίνακας 1.1 Η καταγραφή ενός στιγμιότυπου του προβλήματος των 3 κύβων

Οι τελεστές μετάβασης του προβλήματος μπορούν να περιγραφούν ως εξής:

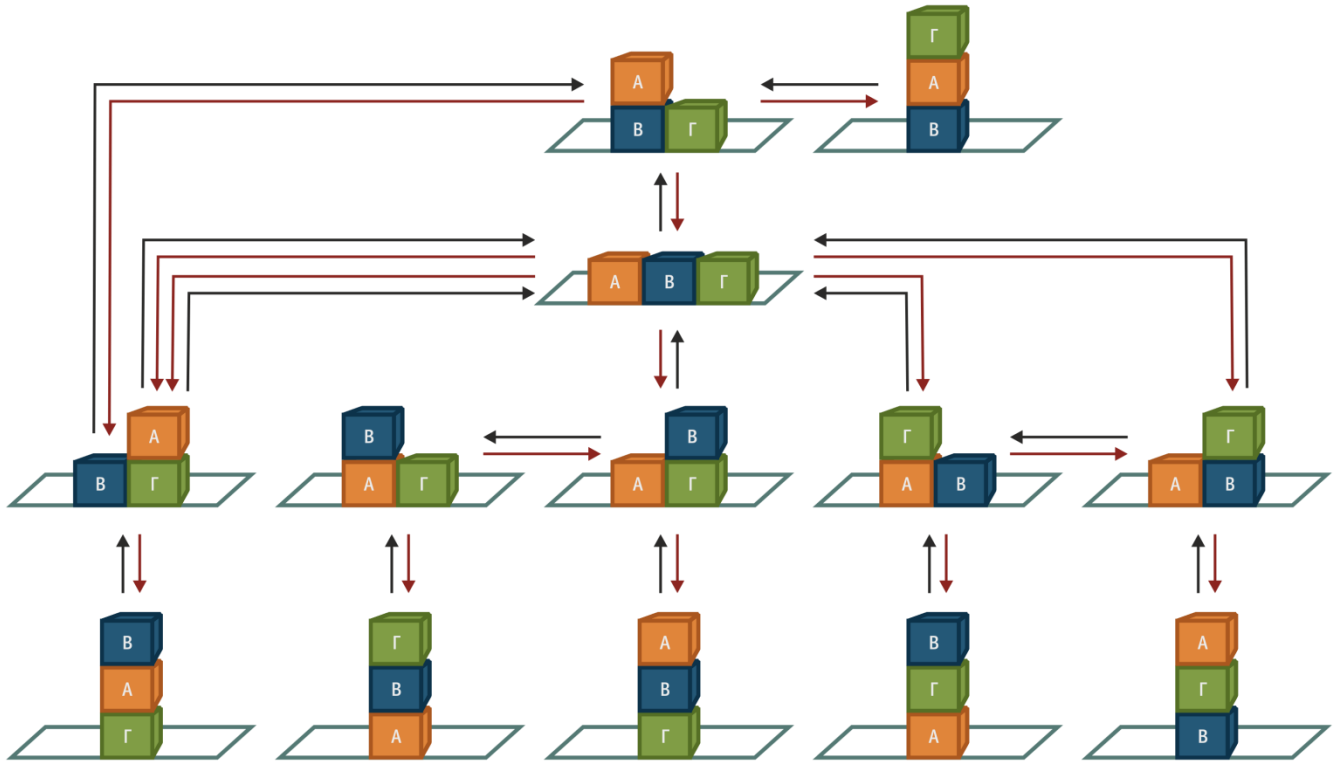
Move-to-cube: Μετακίνησε έναν κύβο x πάνω σε έναν άλλο κύβο y .

Προϋποθέσεις: Οι κύβοι x & y να είναι ελεύθεροι.

Μοντέλο Μετάβασης: Ο κύβος x βρίσκεται πάνω στον κύβο y , ο κύβος y παύει να είναι ελεύθερος.

Move-to-table: Μετακίνησε έναν κύβο x στο τραπέζι.

- Προϋποθέσεις: Ο κύβος x να είναι ελεύθερος και να βρίσκεται πάνω σε έναν άλλο κύβο y .
- Μοντέλο Μετάβασης: Ο κύβος x βρίσκεται πάνω στο τραπέζι, ο κύβος y είναι πλέον ελεύθερος.

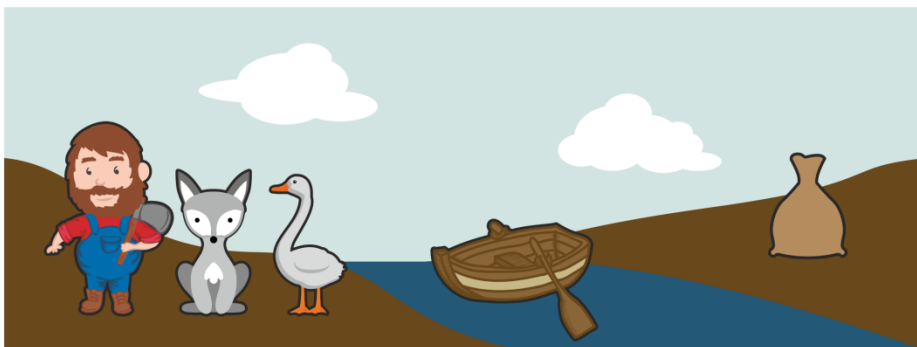


Σχήμα 1.3 Ο χώρος καταστάσεων του προβλήματος των 3 κύβων

Στο σχήμα 1.3 απεικονίζεται ο χώρος καταστάσεων του προβλήματος των 3 κύβων, όπου με κόκκινα βέλη απεικονίζεται ο τελεστής move-to-cube και με μαύρα ο τελεστής move-to-table.

Άλλο πρόβλημα-παιχνίδι είναι ο **αγρότης με τη βάρκα**.

Διατύπωση του προβλήματος του αγρότη με τη βάρκα: Ένας αγρότης (*farmer*) θέλει να περάσει μια χήνα (*goose*), ένα σακί σπόρων (*seeds*) και ένα λύκο (*wolf*) από τη μία όχθη ενός ποταμιού στην άλλη με τη βοήθεια μιας βάρκας. Η βάρκα μπορεί να μεταφέρει τον αγρότη ή μόνο του ή με κάποιο από τους άλλους 3 συντελεστές του προβλήματος, δηλαδή τη χήνα ή το σακί με τους σπόρους ή το λύκο. Ταυτόχρονα, σε καθεμία από τις όχθες δεν μπορούν να μείνουν μόνο τους τα ζεύγη χήνα-σπόροι και λύκος-χήνα, γιατί το ένα τρώει το άλλο.



Εικόνα 1.1 Γραφική αναπαράσταση ενός στιγμιότυπου του προβλήματος του αγρότη με τη βάρκα

Καταστάσεις: Οι έγκυρες καταστάσεις στις οποίες μπορεί να βρεθεί το πρόβλημα φαίνονται στο [σχήμα 1.4](#).

Αρχική κατάσταση: Οποιαδήποτε έγκυρη κατάσταση μπορεί να οριστεί ως αρχική.

Τελεστές μετάβασης: Όλες οι μετακινήσεις που μπορεί να κάνει ο αγρότης από τη μία όχθη στην απέναντι και είναι 4: μετακίνηση του αγρότη μόνου (f), με τη χήνα (g), με το λύκο (w), με τους σπόρους (s). Οι τελεστές επιτρέπεται να εφαρμοστούν σε μια κατάσταση, εφόσον τηρούνται οι περιορισμοί εφαρμογής τους (βλέπε ιδιότητες ακόλουθου πίνακα 1.2).

Στόχος: Μια οποιαδήποτε κατάσταση από το χώρο καταστάσεων μπορεί να τεθεί ως στόχος.

Σε ένα στιγμιότυπο του όπου όλοι βρίσκονται στην αριστερή όχθη, το πρόβλημα του αγρότη με τη βάρκα θα μπορούσε να περιγραφεί με τη βοήθεια του παρακάτω πίνακα:

Αντικείμενα	Ιδιότητες	Σχέσεις
1 αγρότης	Βάρκα δύο ατόμων	Αγρότης στην αριστερή όχθη
1 χήνα	Ο λύκος τρώει τη χήνα	Χήνα στην αριστερή όχθη
1 σακί με σπόρους	αν βρεθούν μόνι.	Σακί με σπόρου στην δεξιά όχθη
1 λύκος	Η χήνα τρώει τους	Λύκος στην αριστερή όχθη
1 βάρκα	σπόρους αν βρεθούν	Βάρκα στην αριστερή όχθη
Αριστερή όχθη	μόνοι.	
Δεξιά όχθη		

Πίνακας 1.2 Περιγραφή μιας κατάστασης του κόσμου του προβλήματος του αγρότη με τη βάρκα

Οι τελεστές μετάβασης του προβλήματος μπορούν να περιγραφούν ως εξής:

f: Μετακίνησε τον αγρότη από την όχθη x στην απέναντι y.

Προϋποθέσεις: Ο αγρότης βρίσκεται στην όχθη x. Στην όχθη x δε βρίσκεται η χήνα με το λύκο ή/και το σακί με τους σπόρους.

Μοντέλο Μετάβασης: Ο αγρότης βρίσκεται στην όχθη y.

g: Μετακίνησε τον αγρότη και τη χήνα από την όχθη x στην απέναντι y.

Προϋποθέσεις: Ο αγρότης και η χήνα βρίσκονται στη όχθη x.

Μοντέλο Μετάβασης: Ο αγρότης και η χήνα βρίσκονται στην όχθη y. .

w: Μετακίνησε τον αγρότη και το λύκο από την όχθη x στην απέναντι y.

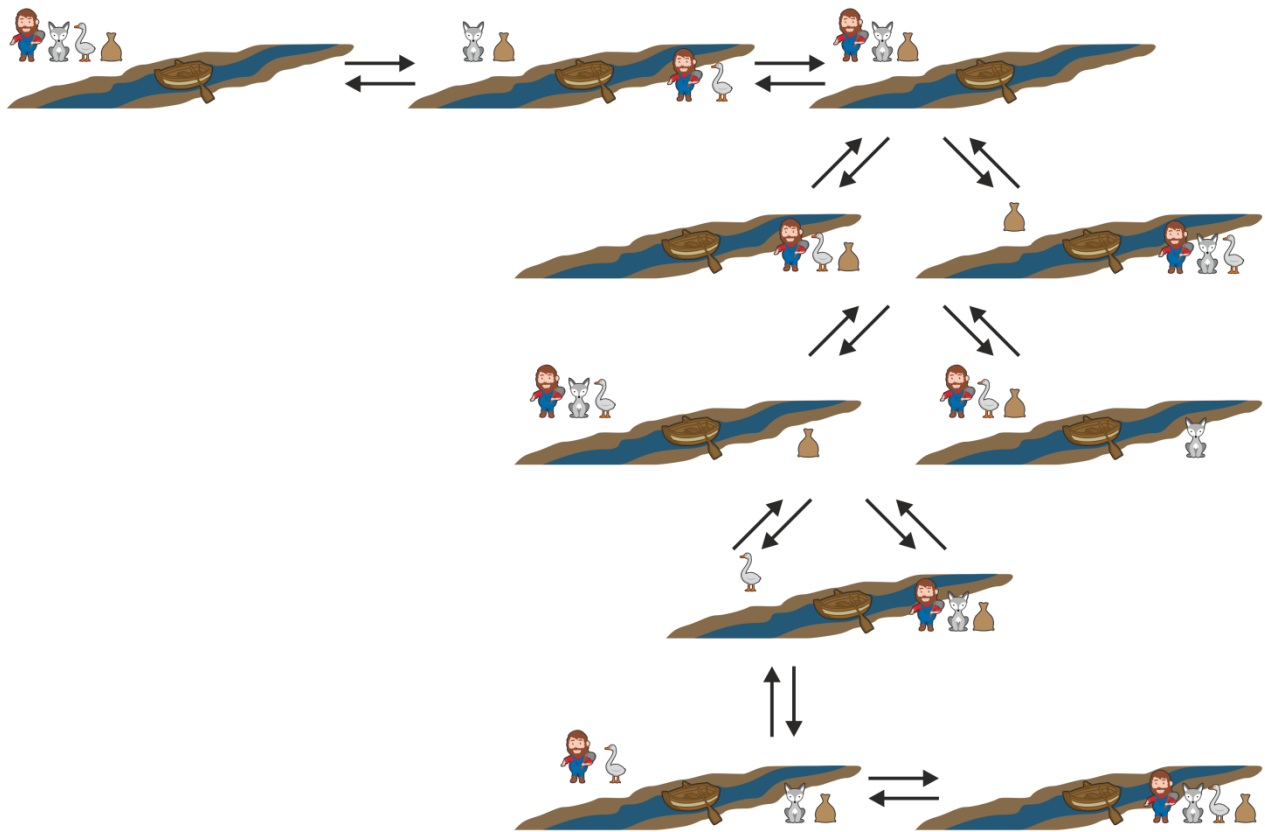
Προϋποθέσεις: Ο αγρότης και ο λύκος βρίσκονται στην όχθη x. Στην όχθη x δε βρίσκεται η χήνα μαζί με τους σπόρους.

Μοντέλο Μετάβασης: Ο αγρότης και ο λύκος βρίσκονται στην όχθη y. .

s: Μετακίνησε τον αγρότη και το σακί με τους σπόρους από την όχθη x στην απέναντι y.

Προϋποθέσεις: Ο αγρότης και το σακί βρίσκονται στην όχθη x. Στην όχθη x δεν βρίσκεται η χήνα μαζί με το λύκο.

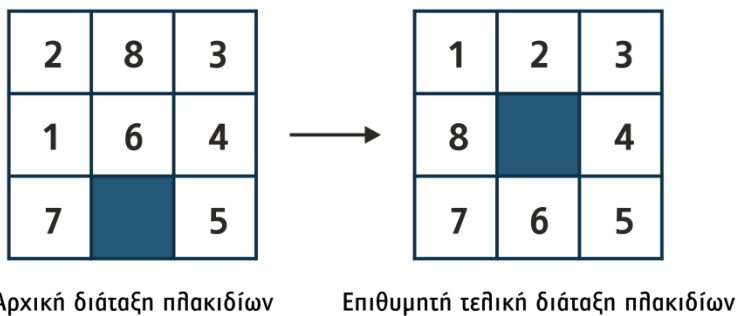
Μοντέλο Μετάβασης: Ο αγρότης και το σακί με τους σπόρους βρίσκονται στην όχθη y. .



Σχήμα 1.4 Ο χώρος καταστάσεων του προβλήματος του αγρότη με τη βάρκα

Επίσης, πολύ γνωστά προβλήματα-παιχνίδια είναι αυτά των **8 πλακιδίων** (8 Puzzles), των 2 δοχείων, των 8 Βασιλισσών, των Πύργων του Ανόι κ.ά.

Διατύπωση του προβλήματος των 8 πλακιδίων (8 Puzzles): Ένα πλαίσιο 3x3 περιέχει 8 πλακίδια, αριθμημένα από το 1 έως το 8, και μία θέση κενή χωρίς πλακίδιο. Τα πλακίδια μπορούν να μετακινηθούν σε μια γειτονική θέση, εφόσον αυτή η θέση είναι κενή. Σκοπός είναι να βρεθούν τα πλακίδια αυτά σε μια επιθυμητή διάταξη. Το πρόβλημα μπορεί να γενικευτεί για N πλακίδια.



Σχήμα 1.5 Το πρόβλημα των 8 πλακιδίων

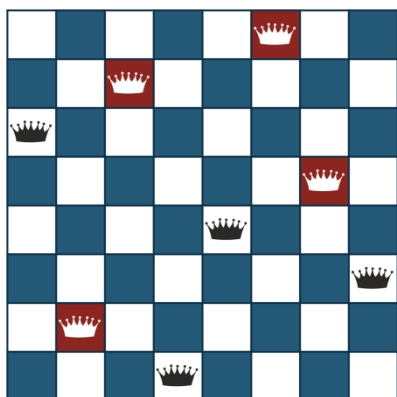
Διατύπωση προβλήματος των 2 δοχείων: Στο πρόβλημα υπάρχουν δυο δοχεία, που χωράνε 4 λίτρα και 3 λίτρα νερού αντίστοιχα, και μια βρύση που μπορεί να τα γεμίσει πλήρως είτε είναι άδεια είτε μισογεμάτα. Το ζητούμενο είναι να βρεθούν μέσα στο μικρό δοχείο ακριβώς 2 λίτρα νερό. Επιτρεπτές

ενέργειες είναι να γεμίσει ένα δοχείο πλήρως από τη βρύση, να αδειάσει τελείως ένα δοχείο από το περιεχόμενό του και να μεταγγιστεί το περιεχόμενο του ενός δοχείου στο άλλο, αν χωράει όλο, αλλιώς το μέρος που χωράει.



Σχήμα 1.6 Το πρόβλημα των 2 δοχείων

Διατύπωση προβλήματος των 8 βασίλισσών: Το πρόβλημα των 8 βασίλισσών είναι παιχνίδι στρατηγικής, όπου σε μια σκακιέρα πρέπει να τοποθετηθούν 8 βασίλισσες έτσι, ώστε να μην απειλεί η μια την άλλη, δηλαδή καμιά να μη μοιράζεται την ίδια γραμμή, στήλη ή διαγώνιο με τις υπόλοιπες. Μπορεί να λυθεί με αναδρομή και άλλες κλασικές προσεγγίσεις, με αλγόριθμους αναζήτησης, αλλά και με μη παραδοσιακές προσεγγίσεις, όπως οι γενετικοί αλγόριθμοι.



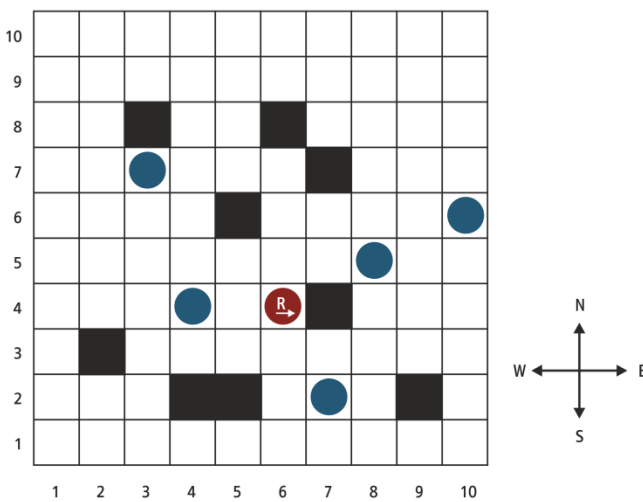
Σχήμα 1.7 Το πρόβλημα των 8 βασίλισσών

Διατύπωση προβλήματος των Πύργων του Ανόι: Δίνεται ένας αριθμός κρίκων διαφορετικού μεγέθους και τρεις στύλοι. Οι κρίκοι είναι τοποθετημένοι στον πρώτο στύλο με διάταξη από τον μεγαλύτερο χαμηλότερα προς τον μικρότερο στην κορυφή. Από το στύλο όπου βρίσκονται οι κρίκοι πρέπει να μεταφερθούν ένας με ίδια τελική διάταξη στον τρίτο στύλο, με τη βοήθεια του μεσαίου στύλου, χωρίς ποτέ ένας μεγαλύτερος κρίκος να τοποθετηθεί πάνω από έναν μικρότερο. Συνήθως λύνεται με αναδρομή αλλά και με κλασσικές μεθόδους αναζήτησης.



Σχήμα 1.8 Σχηματική περιγραφή των Πύργων του Ανόι

Διατύπωση προβλήματος του ρομπότ: Ένα ρομπότ κινείται κατά μία θέση οριζοντίως ή καθέτως σε ένα χώρο με εμπόδια. Σε διάφορες θέσεις υπάρχουν αντικείμενα τα οποία προσπαθεί να εντοπίσει το ρομπότ. Στόχος είναι το ρομπότ να καταφέρει να εντοπίσει όλα τα αντικείμενα ή κάποια από αυτά. Στην κατηγορία αυτή ανήκει το παιχνίδι του πάκμαν και γενικότερα παιχνίδια που χαρακτηρίζονται ως λαβύρινθοι.



Σχήμα 1.9 Το πρόβλημα του ρομπότ

Προβλήματα πραγματικού κόσμου

Τα προβλήματα πραγματικού κόσμου είναι κατά κανόνα δύσκολο να οριστούν. Κλασικότεροι εκπρόσωποί τους είναι τα προβλήματα περιήγησης, μεταξύ των οποίων συγκαταλέγεται το γνωστό **πρόβλημα του περιοδεύοντος πωλητή** (travelling salesperson problem-TSP), στο οποίο έχουμε αναφερθεί στην αρχή του Κεφαλαίου (βλέπε [Σχήμα 1.1](#)). Άλλα προβλήματα είναι τα προβλήματα διάταξης κυκλώματος, σχεδίασης πρωτεϊνών, κίνησης ρομπότ, αναζήτησης στο διαδίκτυο κτλ.

1.1.5 Λύση προβλήματος

Λύση ενός προβλήματος P , όπου $P=(I,G,T,S)$, είναι μια ακολουθία από τελεστές μετάβασης t_1, t_2, \dots, t_n , με την ιδιότητα:

$$G = t_n (\dots (t_2 (t_1 (I))) \dots)$$

Παράδειγμα: Αν το πρόβλημα του αγρότη οριστεί τυπικά ως ακολούθως:

I (input)= Αγρότης-Λύκος-Χήνα-Σακί με σπόρους στην αριστερή όχθη
 G (goal)=Αγρότης-Λύκος-Χήνα-Σακί στη δεξιά όχθη
 T (transitions)={

- F**(armer) : Μετάφερε τον αγρότη από την όχθη όπου βρίσκεται στην απέναντι.
 - W**(olf) : Μετάφερε τον αγρότη και το λύκο από την όχθη όπου βρίσκονται στην απέναντι.
 - G**(oose) : Μετάφερε τον αγρότη και τη χήνα από την όχθη όπου βρίσκονται στην απέναντι.
 - S**(eeds) : Μετάφερε τον αγρότη και το σακί από την όχθη όπου βρίσκονται στην απέναντι.
- }

Αν ο χώρος καταστάσεων S του προβλήματος είναι αυτός του [σχήματος 1.4](#), τότε, υπάρχουν δύο εναλλακτικές λύσεις που είναι οι ακόλουθες:

$$G \rightarrow F \rightarrow S \rightarrow G \rightarrow W \rightarrow F \rightarrow G$$

$$G \rightarrow F \rightarrow W \rightarrow G \rightarrow S \rightarrow F \rightarrow G$$

Εναλλακτικός ορισμός είναι ο ακόλουθος:

Λύση ενός προβλήματος είναι μια ακολουθία από καταστάσεις στις οποίες μπορεί να βρεθεί διαδοχικά το πρόβλημα, με αρχή από την αρχική κατάσταση και κατάληξη σε μία από τις επιθυμητές τελικές καταστάσεις.

Ως **μερική λύση** (partial solution) εννοούμε μια ακολουθία βημάτων επίλυσης που δεν έχουν καταλήξει ακόμα σε επιτυχία ή αποτυχία: παράδειγμα, στο πρόβλημα του αγρότη τρεις χρονικά διαδοχικές μερικές λύσεις θεωρούνται οι:

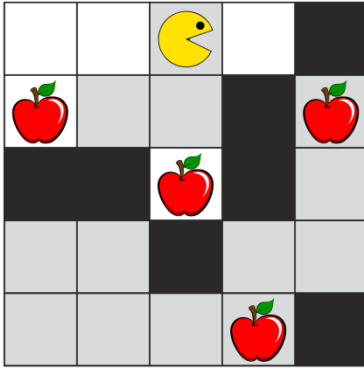
$$G \rightarrow F \rightarrow S$$

$$G \rightarrow F \rightarrow S \rightarrow G$$

$$G \rightarrow F \rightarrow S \rightarrow G \rightarrow W$$

Ένα πρόβλημα μπορεί να έχει πολλές λύσεις που προκύπτουν κατά την αναζήτηση μέσα στο χώρο καταστάσεων. Δοθέντος ενός προβλήματος $P = (I, G, T, S)$, ο **χώρος αναζήτησης** (search space) είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση K. Μια κατάσταση K_i χαρακτηρίζεται ως **προσβάσιμη**, αν υπάρχει μια ακολουθία τελεστών μετάβασης που οδηγεί από την αρχική κατάσταση K στην κατάσταση K_i .

Ο χώρος αναζήτησης ενός προβλήματος άλλοτε ταυτίζεται με το χώρο καταστάσεων και άλλοτε, για πολύ μεγάλους χώρους καταστάσεων ή για συγκεκριμένου τύπου προβλήματα, περιορίζεται ως υποσύνολο του σε ένα μικρότερο ή μεγαλύτερο μέρος του. Για παράδειγμα, στο πρόβλημα του αγρότη οι δύο χώροι ταυτίζονται, ενώ στην περίπτωση του προβλήματος του πάκμαν, που μπορεί να θεωρηθεί ως παραλλαγή του προβλήματος του ρομπότ, διαφοροποιούνται, αν ο κόσμος του προβλήματος στο αρχικό του στιγμιότυπο είναι αυτός που δίνεται στο σχήμα 1.10 (με γκρι σημειώνονται οι καταστάσεις του χώρου καταστάσεων στις οποίες δεν μπορεί να βρεθεί το πάκμαν κατά την αναζήτηση, δεδομένου ότι δεν μπορεί να τις προσεγγίσει λόγω των εμποδίων).



Σχήμα 1.10 Γραφική απεικόνιση του χώρου αναζήτησης του προβλήματος του πάκμαν

Ένας χώρος αναζήτησης μπορεί να αναπαρασταθεί ως γράφημα με κόμβους τις προσβάσιμες καταστάσεις του γράφου του χώρου καταστάσεων.

1.2 Αναζήτηση λύσης

Όταν ο χώρος καταστάσεων ενός προβλήματος είναι πολύ μεγάλος, τότε κρίνεται αναγκαία η ύπαρξη μιας συστηματικής μεθόδου έρευνάς του, δηλαδή ενός **αλγόριθμου αναζήτησης** (search algorithm). Με συνδυασμό και επέκταση των κλασικών αλγορίθμων αναζήτησης, τους οποίους και θα παρουσιάσουμε στις επόμενες παραγράφους, μπορούν να προκύψουν πολλοί ακόμα αλγόριθμοι κατάλληλοι για την επίλυση προβλημάτων συγκεκριμένων τύπων. Οι αλγόριθμοι αναζήτησης περιορίζονται στο χώρο αναζήτησης και χαρακτηρίζονται από την πολυπλοκότητα και την αποδοτικότητά τους.

Η αναζήτηση λύσης επιτυγχάνεται σε κύκλους αναζήτησης. Σε κάθε κύκλο επεκτείνεται το σύνολο των μερικών λύσεων που έχουν σχηματιστεί κατά τον προηγούμενο κύκλο. Η επιλογή του τρόπου επέκτασης εξαρτάται από τη στρατηγική αναζήτησης που έχει υιοθετήσει ο αλγόριθμος αναζήτησης.

Κατά τη διαδικασία αναζήτησης, αναπτύσσεται το δέντρο αναζήτησης που περιλαμβάνει μέρος ή το σύνολο των καταστάσεων του χώρου αναζήτησης.

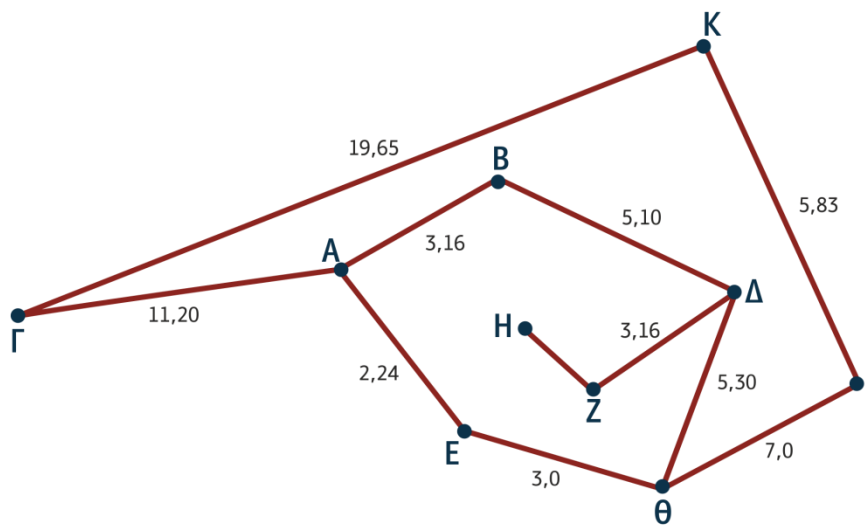
1.2.1 Δένδρο Αναζήτησης

Βασικό εργαλείο για την εφαρμογή ενός αλγορίθμου αναζήτησης μέσα στο χώρο καταστάσεων είναι το **δένδρο αναζήτησης** (search tree). Το δένδρο αναζήτησης σχηματίζεται δυναμικά, καθώς ο αλγόριθμος ανιχνεύει όλες τις δυνατές να σχηματιστούν ακολουθίες διαδοχικών καταστάσεων που μπορεί να οδηγήσουν σε λύση. Κάθε τέτοια ακολουθία καταστάσεων, η οποία ξεκινά από την αρχική κατάσταση και καταλήγει σε ένα φύλλο του μέχρι εκείνου του σημείου ανεπτυγμένου δένδρου, καλείται **μονοπάτι αναζήτησης** (search path). Το σύνολο των αποτυπωμένων πάνω σε ένα δένδρο αναζήτησης εναλλακτικών μονοπατιών σε μια συγκεκριμένη στιγμή της εξέλιξης του κόσμου του προβλήματος καλείται **ουρά αναζήτησης** (search queue ή fringe). Τα μονοπάτια που αποτυπώνονται σε ένα δένδρο μιας μη ολοκληρωμένης αναζήτησης μπορούν να θεωρηθούν μερικές λύσεις του προβλήματος.

Σε ένα δένδρο αναζήτησης που αντιστοιχεί σε ένα λυμένο πρόβλημα, μια λύση του αποτελεί ένα ολοκληρωμένο μονοπάτι που αποδίδεται ως μια ακολουθία διαδοχικών καταστάσεων αρχίζοντας από την αρχική κατάσταση και καταλήγοντας σε μια τελική. Είναι σημαντικό να διαφοροποιηθούν οι έννοιες δέντρο αναζήτησης, χώρος αναζήτησης και χώρος καταστάσεων.

Ένα από τα γνωστότερα προβλήματα στο χώρο της ΤΝ, όπως προαναφέραμε, είναι το πρόβλημα του περιοδεύοντος πωλητή, το οποίο στο κεφάλαιο αυτό θα απλοποιήσουμε, για να καταστεί πρόβλημα-υπόδειγμα για την παρουσίαση των διαφορετικών αλγορίθμων αναζήτησης. Στην απλουστευμένη αυτή εκδοχή, ο πωλητής δεν έχει ως στόχο να επιστρέψει απαραίτητα στην αρχική πόλη, αλλά σε κάποια οποιαδήποτε από τις πόλεις του προβλήματος, χωρίς να απαιτείται να έχει επισκεφθεί όλες τις υπόλοιπες. Το

πρόβλημα θεωρείται λυμένο, όταν βρεθεί ένα οποιοδήποτε μονοπάτι που να οδηγεί από την αρχική πόλη στην πόλη-στόχο. Στο παρακάτω σχήμα δίνεται η περιγραφή μιας εκδοχής του χώρου αναζήτησης του περιοδεύοντος πωλητή, όταν στο πρόβλημα υπάρχουν 10 πόλεις που συνδέονται μεταξύ τους με δρόμους διαφορετικού μήκους. Στην περιγραφή αυτή ο κάθε κόμβος αντιστοιχεί σε μια πόλη του προβλήματος και κάθε ακμή μεταξύ των κόμβων στους υπάρχοντες δρόμους. Επιπλέον, σε κάθε ακμή σημειώνεται η απόσταση μεταξύ των πόλεων που συνδέει.

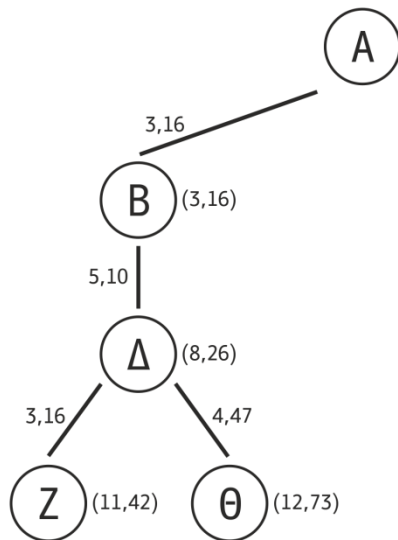


Σχήμα 1.11 Μια εκδοχή του χώρου αναζήτησης του προβλήματος του περιοδεύοντος πωλητή, όταν επισκέπτεται 10 πόλεις

Ο ορισμός του συγκεκριμένου προβλήματος έχει ως εξής:

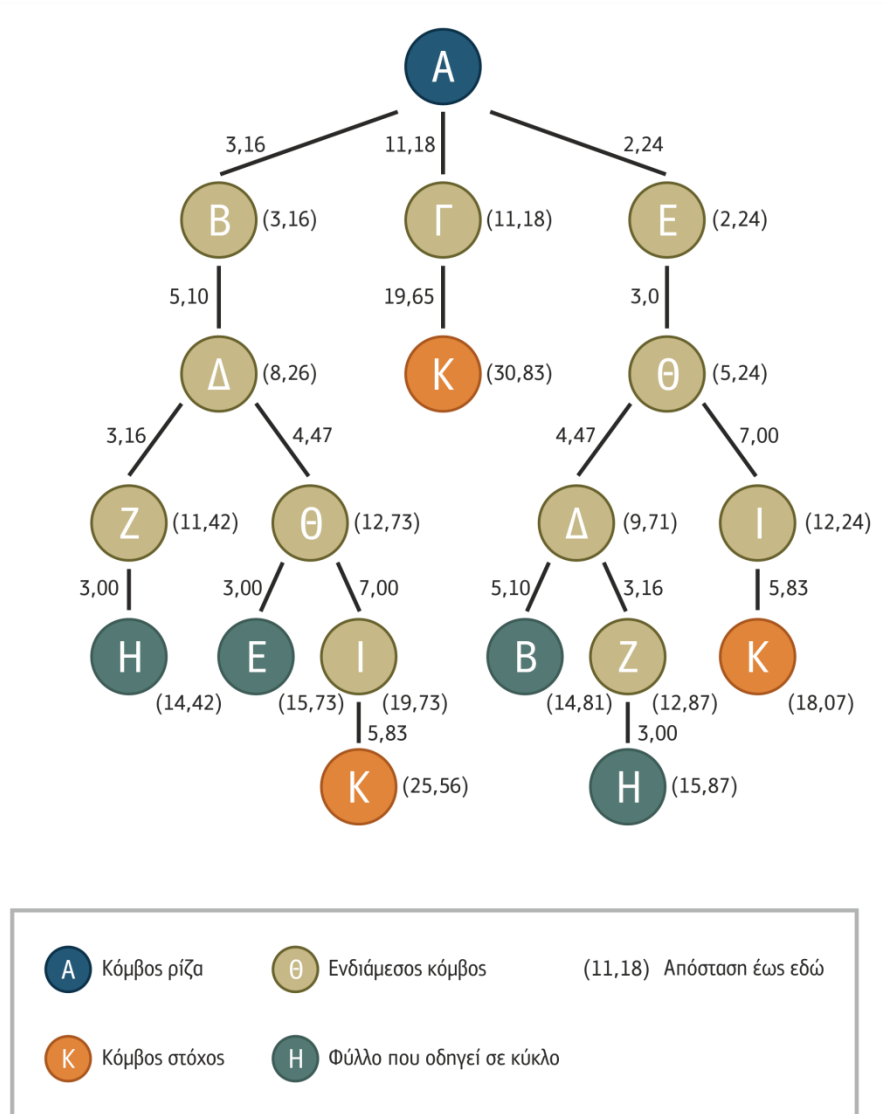
- I= Πωλητής σε πόλη A
- G= Πωλητής σε πόλη K
- T={
 Μετάφερε τον Πωλητή από πόλη x σε πόλη y
 }
- S= κόμβοι του γραφήματος του σχήματος 1.11

Οι αλγόριθμοι αναζήτησης εξερευνούν γραφήματα όπως αυτόν στο σχήμα 1.11 χρησιμοποιώντας και εμπλουτίζοντας κατά την πορεία προς εύρεση λύσης τις γνώσεις σχετικά με τη δυνατότητα σύνδεσης απομακρυσμένων κόμβων μεταξύ τους και το μήκος των εναλλακτικών μονοπατιών που τις συνδέουν. Στην προκειμένη περίπτωση, ένας τύπος γνώσης που παράγεται βήμα-βήμα είναι αυτή που αφορά την απόσταση κάθε κόμβου από τον κόμβο-ρίζα. Για παράδειγμα, σε ένα στιγμιότυπο της επίλυσης του προβλήματος του πωλητή, όταν αυτός ξεκινά από την πόλη A, ο αλγόριθμος θα μπορούσε να έχει διανύσει το μονοπάτι A-B-Δ-Θ και να έχει αποκτήσει τη γνώση ότι η πόλη A συνδέεται με τη Θ με μεταξύ τους απόσταση 12,73 χιλ. (βλέπε Σχήμα 1.12).



Σχήμα 1.12 Απόσταση απομακρυσμένων κόμβων ενός γραφήματος

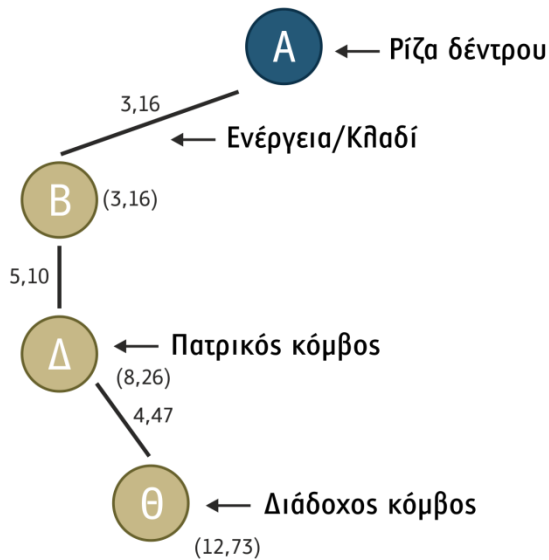
Όταν εκτελείται **εξαντλητική αναζήτηση** (exhaustive search), δηλαδή ο αλγόριθμος προσπαθεί να βρει όλες τις δυνατές λύσεις του προβλήματος, τότε στο δένδρο που σχηματίζεται αποτυπώνονται όλα τα μονοπάτια που είναι δυνατόν να διανυθούν (μερικά από τα οποία θα αποδειχθούν λύσεις), καθώς και όλη η γνώση που αποκτάται κατά τη διάνυση αυτή (βλέπε Σχήμα 1.13).



Σχήμα 1.13 Το δένδρο της εξαντλητικής αναζήτησης του προβλήματος του περιοδεύοντος πωλητή

1.2.2 Περιγραφή Δένδρων Αναζήτησης

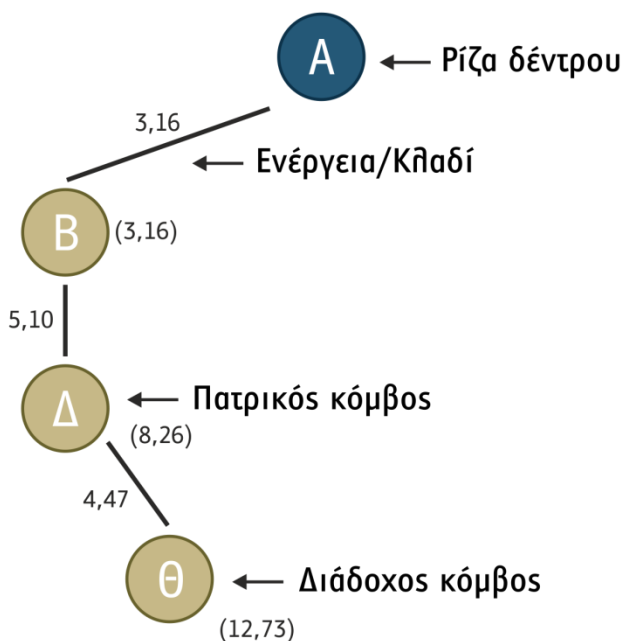
Κάθε κόμβος του δέντρου αναζήτησης αντιστοιχεί σε μια κατάσταση του χώρου καταστάσεων του προβλήματος. Η ρίζα του δένδρου αντιστοιχεί στην αρχική κατάσταση. Ο κόμβος που δημιουργεί ένα συγκεκριμένο κόμβο καλείται **πατρικός** ή **γονικός κόμβος** (parent node). Οι κόμβοι που δημιουργούνται από ένα γονικό κόμβο καλούνται **διάδοχοι κόμβοι** (successor nodes). Η ενέργεια που εφαρμόζεται σε έναν κόμβο, για να καταστεί γονικός κόμβος, προκαλείται από κάποιον από τους τελεστές του προβλήματος και παριστάνεται ως ακμή ή κλαδί του δένδρου που ενώνει το γονικό κόμβο με καθέναν από τους διάδοχους κόμβους που δημιουργεί. Κάθε κόμβος του δένδρου αντιστοιχεί στην κατάσταση στην οποία καταλήγει το μονοπάτι που προκύπτει από την κατά ένα βήμα επέκταση του μονοπατιού που έχει ήδη καθορίσει ο γονικός του κόμβος. Το κόστος του αλγόριθμου, για να διανύσει ένα μονοπάτι, καλείται **κόστος μονοπατιού** (path cost). Το πλήθος των κόμβων μέσα σε ένα μονοπάτι καλείται **βάθος μονοπατιού** (path depth). Για παράδειγμα, στο μονοπάτι του σχήματος 1.14 το κόστος είναι 12,73 και το βάθος είναι 4.



Σχήμα 1.14 Μονοπάτι βάθους 4 και κόστους 12,73 για το πρόβλημα του περιοδεύοντος πωλητή

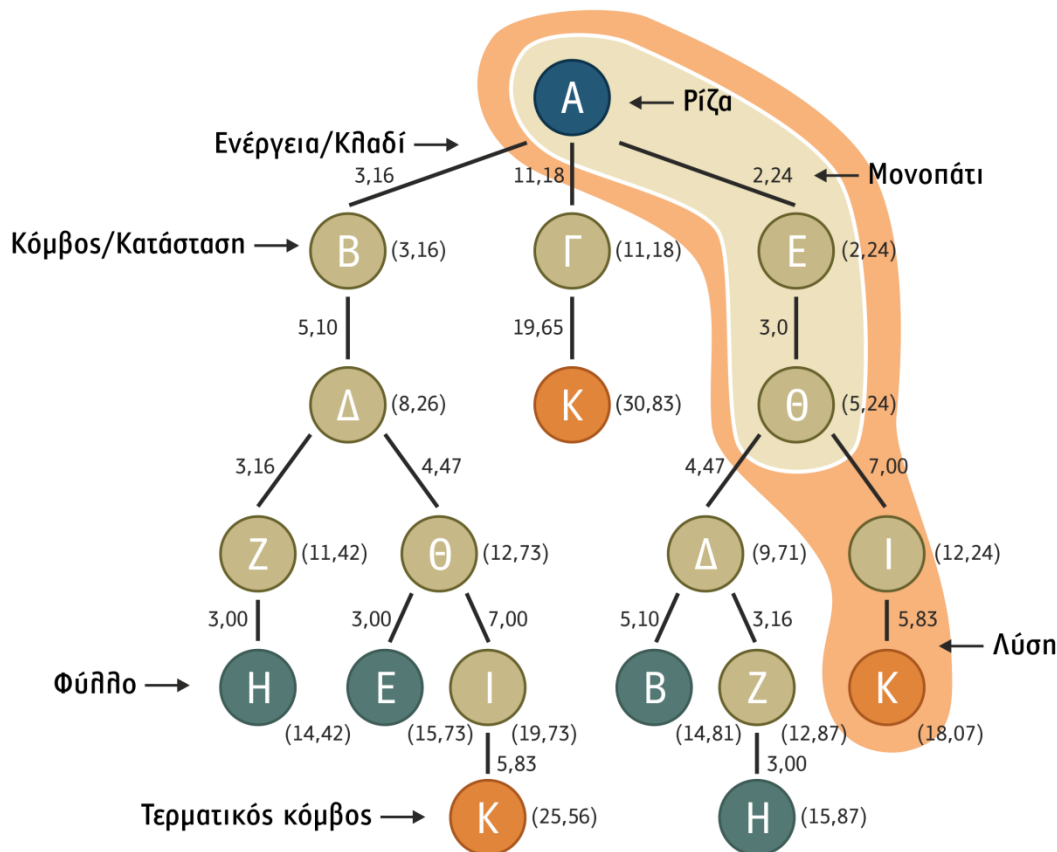
Μέτωπο (fringe ή front) καλείται το διατεταγμένο σύνολο των κόμβων που ο αλγόριθμος έχει αναρτήσει στο δένδρο, αλλά δεν έχει ακόμα επεκτείνει προς διάδοχους κόμβους, δηλαδή τα φύλλα του δένδρου σε κάποιο στιγμιότυπο της εκτέλεσης του αλγόριθμου. Στο μέτωπο δεν περιλαμβάνονται οι κόμβοι που έχουν ήδη γίνει γονικοί κόμβοι, ακόμα και αν αυτοί αποτελούν φύλλα του δένδρου (περίπτωση μονοπατιών των οποίων η επέκταση θα δημιουργούσε κύκλους): για παράδειγμα, η περίπτωση του κόμβου Η στο δένδρο του σχήματος 1.15 καλείται και κατάσταση αδιεξόδου. Στο ίδιο σχήμα, το μέτωπο περιλαμβάνει τους κόμβους Θ, Γ και Ε.

Το σύνολο των κόμβων ενός δένδρου αναζήτησης που ο αλγόριθμος έχει ήδη καταστήσει γονικούς καλείται **κλειστό σύνολο** (closed set). Στο σχήμα 1.15, το κλειστό σύνολο περιλαμβάνει τους κόμβους Α, Β, Δ, Ζ, Η, δηλαδή όλους τους κόμβους που δεν ανήκουν στο μέτωπο.



Σχήμα 1.15 Το δένδρο αναζήτησης του προβλήματος του περιοδεύοντος πωλητή σε ένα στιγμιότυπο της εξέλιξής του

Όλα όσα αναφέραμε σχετικά με τα δένδρα αναζήτησης έως τώρα αποτυπώνονται συγκεντρωτικά στο σχήμα 1.16:



Σχήμα 1.16 Ανάλυση δένδρου αναζήτησης

1.3 Αλγόριθμοι Αναζήτησης

Για την επίλυση ενός προβλήματος πρέπει να αποφασιστεί ποιος από τους υπάρχοντες αλγόριθμους θα επιλεγεί. Αυτή η απόφαση θα χαρακτηρίσει την πορεία επίλυσης και θα καθορίσει αποφασιστικά την ποιότητα της λύσης ή των λύσεων που θα βρεθούν.

Τα βασικά στοιχεία που χαρακτηρίζουν μια αναζήτηση είναι τα παρακάτω:

- ο αλγόριθμος που χρησιμοποιήθηκε (A),
- το σύνολο των καταστάσεων που εξετάστηκαν (V),
- το σύνολο των λύσεων που βρέθηκαν (F),
- το σύνολο των τελικών καταστάσεων που εξετάστηκαν (GS).

Βάσει των παραπάνω, η επίλυση ενός προβλήματος Ps (problem solution) μέσω ενός αλγόριθμου αναζήτησης A μπορεί να παρουσιαστεί αναλυτικά ως ακολούθως:

$$Ps = (V, A, F, GS)$$

Ένας αλγόριθμος δεν είναι βέβαιο ότι θα βρει πάντα τη λύση σε ένα πρόβλημα, ακόμα και αν αυτή υπάρχει. Ένας αλγόριθμος καλείται *πλήρης* (complete), όταν εγγυάται την εύρεση λύσης σε περίπτωση κατά την οποία αυτή υπάρχει. Σε αντίθετη περίπτωση καλείται *ατελής* (incomplete).

Τα κριτήρια επιλογής ενός αλγορίθμου αναζήτησης είναι τα ακόλουθα:

- η πληρότητα ή ατέλεια,
- η ικανότητα ή μη για εξαντλητική αναζήτηση,
- η ποιότητα των λύσεων που βρίσκει (αν βρίσκει τη βέλτιστη λύση πρώτη),
- η αποδοτικότητά του σε χρόνο,
- η αποδοτικότητά του σε μνήμη,
- η ευκολία υλοποίησής του.

Για να εκτιμήσουμε την αποδοτικότητα ενός αλγορίθμου αναζήτησης, εξετάζουμε:

- το κόστος αναζήτησης (που εξαρτάται από την πολυπλοκότητα) ή και
- το κόστος λύσης (δηλαδή το κόστος σε μνήμη της διαδρομής που βρέθηκε ως λύση)

Σε περίπτωση κατά την οποία μας ενδιαφέρουν και τα δύο, εξετάζουμε το ολικό κόστος. Η βαρύτητα που προσδίδουμε στο κάθε κόστος εξαρτάται από την εφαρμογή. Άλλοτε θέλουμε μια πολύ καλή (ή και τη βέλτιστη) λύση, χωρίς να μας απασχολεί ο χρόνος που θα αφιερώσουμε, για να τη βρούμε, και άλλοτε θέλουμε μια (οποιαδήποτε) λύση που θα βρεθεί γρήγορα.

1.3.1 Γενικός αλγόριθμος αναζήτησης

Για τη γενική διαδικασία αναζήτησης λύσης που εφαρμόζει κάθε αλγόριθμος μπορούμε να σημειώσουμε ότι η αναζήτηση:

- επιτυγχάνεται μέσα στο χώρο καταστάσεων.
- αρχίζει από ένα αρχικό μέτωπο το οποίο περιέχει μια μοναδική κατάσταση, την αρχική.
- αρχίζει με μια μερική λύση, που, όταν περιγράφεται με καταστάσεις, τότε έχει τη μορφή ενός μονοπατιού μήκους 1 που περιέχει μόνο την αρχική κατάσταση.
- ως διαδικασία λειτουργεί σε κύκλους και βασίζεται στη συγκέντρωση των μερικών λύσεων μέσα στην ουρά και την επέκτασή τους.

Πιο αναλυτικά, ο γενικός αλγόριθμος αναζήτησης δομεί το δένδρο αναζήτησης ξεκινώντας από την αρχική κατάσταση του προβλήματος, που αποτελεί τη ρίζα του δένδρου, και καταλήγοντας σε κόμβους-φύλλα, μερικά από τα οποία μπορεί να είναι αδιέξοδα και άλλα να είναι στόχοι.

Καταρχάς, δημιουργείται το αρχικό μονοπάτι ως ένα μονοπάτι μηδενικού μήκους που περιέχει μόνο τη ρίζα του δένδρου (αρχική κατάσταση) και, παράλληλα, δημιουργείται η αρχική ουρά, που περιέχει μόνο το αρχικό μονοπάτι. Στη συνέχεια, δημιουργούνται νέα μονοπάτια στην ουρά, καθώς οι υπάρχοντες κόμβοι γίνονται γονικοί κόμβοι. Το σύνολο των κόμβων στους οποίους καταλήγουν τα μονοπάτια της ουράς, σε ένα δεδομένο στιγμιότυπο της αναζήτησης, δημιουργούν το τρέχον μέτωπο αναζήτησης.

Για να γίνουν πιο κατανοητά τα παραπάνω, θα παρουσιάσουμε πώς θα τα υλοποιούσε οποιοσδήποτε αλγόριθμος αναζήτησης στον πρώτο κύκλο λειτουργίας του κατά την επίλυση του προβλήματος του αγρότη με τις 10 πόλεις (βλέπε [Σχήμα 1.11](#)).

Αρχικοποίηση δένδρου αναζήτησης:

Αρχικό μέτωπο: <A>
Αρχικό μονοπάτι: (A)
Αρχική ουρά: <(A)>
Αρχικό κλειστό σύνολο: {}

1ος κύκλος λειτουργίας:

Μέτωπο: <B, Γ, E>
Ουρά: <(A B), (A Γ), (A E)>

Κλειστό: {A}

όπου τα σύμβολα '<' και '>' χρησιμοποιούνται, για να περιγραφεί ένα διατεταγμένο σύνολο από καταστάσεις ή μονοπάτια και τα '(' και ')' χρησιμοποιούνται, για να περιγραφεί μια ακολουθία από καταστάσεις που περιγράφει ένα μονοπάτι.

Ο τρόπος επέκτασης στον επόμενο 2ο κύκλο διαφέρει από αλγόριθμο σε αλγόριθμο, ανάλογα με τον κόμβο του μετώπου που ο κάθε αλγόριθμος θα επιλέξει ως γονικό κόμβο. Αν η αναζήτηση χρησιμοποιούσε την πρώτη σε βάθος αναζήτηση που θα περιγραφεί [σε επόμενη παράγραφο](#), τότε θα επιλέγετο ως γονικός κόμβος ο B με διάδοχο κόμβο τον Δ και θα ίσχυαν τα επόμενα:

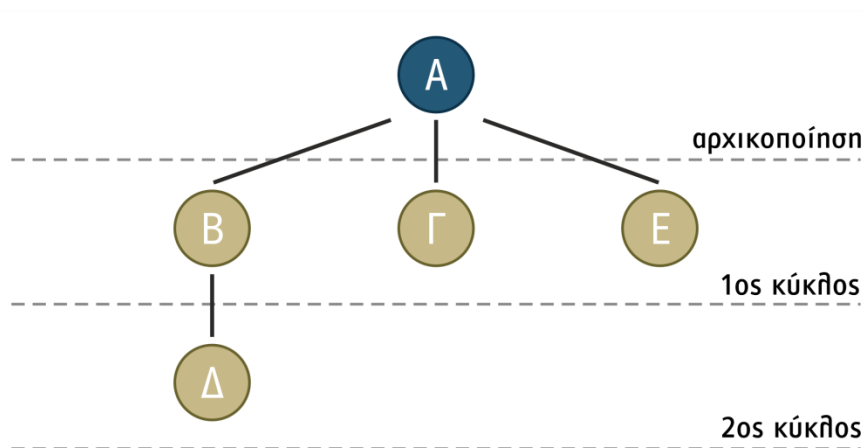
2ος κύκλος λειτουργίας:

Μέτωπο: <Δ, Γ, Ε>

Ουρά: <(A B Δ), (A Γ), (A Ε)>

Κλειστό: {A, B}

Τα παραπάνω αποτυπώνονται στο σχήμα 1.17 που ακολουθεί.



Σχήμα 1.17 Ανάπτυξη δένδρου αναζήτησης για το πρόβλημα του αγρότη έως τον 2^ο κύκλο λειτουργίας

Δείτε [κινούμενη εικόνα 1.1 - Ανάπτυξη δένδρου Αναζήτησης σε κύκλους](#)

Η διαδικασία αναζήτησης θα μπορούσε να παρουσιαστεί και ως μια διαδικασία επιλογής γονικού κόμβου από το μέτωπο αναζήτησης και επέκτασης του μονοπατιού που καταλήγει σε αυτόν προς τους διάδοχους κόμβους ή κόμβους-παιδιά. Οι αλγόριθμοι αναζήτησης χαρακτηρίζονται από τις στρατηγικές που εφαρμόζουν, για να επιλέξουν έναν από τους κόμβους του μετώπου ως γονικό κόμβο, αλλά όλοι ακολουθούν ορισμένα κοινά βήματα που συγκροτούν έναν γενικό αλγόριθμο αναζήτησης.

Τα βήματα του γενικού αλγόριθμου αναζήτησης είναι τα ακόλουθα:

Βήματα Γενικού Αλγόριθμου Αναζήτησης

1. Βάλτε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης.
2. Αν το μέτωπο είναι κενό, τότε η αναζήτηση ολοκληρώθηκε ανεπιτυχώς, χωρίς να βρεθεί λύση, ΤΕΛΟΣ.
3. Αν ο πρώτος σε σειρά κόμβος του μετώπου είναι στόχος, τότε βρέθηκε λύση, ΤΕΛΟΣ.
4. Αν ο πρώτος σε σειρά κόμβος του μετώπου ανήκει στο κλειστό σύνολο, τότε Πήγαινε στο βήμα 2.

5. Εφάρμοσε τους τελεστές στον πρώτο σε σειρά κόμβο του μετώπου, για να γίνει γονικός κόμβος και να παραχθούν οι διάδοχοι κόμβοι.
6. Πρόσθεσε τους διάδοχους κόμβους στο μέτωπο, αφού αφαιρέσεις το γονικό κόμβο, και δημιούργησε το νέο μέτωπο.
7. Αν το απαιτεί ο αλγόριθμος αναζήτησης, αναδιάταξε τους κόμβους του μετώπου σύμφωνα με το ποιοτικό κριτήριο που ορίζει η στρατηγική του αλγόριθμου.
8. Βάλε το γονικό κόμβο στο κλειστό σύνολο.
9. Πήγαινε στο βήμα 2.

Σε κάθε κύκλο λειτουργίας ενός αλγόριθμου αναζήτησης, το βήμα 5 ορίζει τον πρώτο κόμβο του μετώπου ως γονικό, αλλά τα βήματα 6 και 7 όρισαν το ποιος κόμβος θα βρεθεί στην πρώτη θέση .

Ο αλγόριθμος αναζήτησης αναπτύσσεται ως μια συνάρτηση που καλεί επαναληπτικά άλλες συναρτήσεις, για να υλοποιήσει τα παραπάνω βήματα.

Δείτε [ρομπρ 1.1 - Ψευδοκώδικας Γενικού Αλγόριθμου Αναζήτησης με παρακολούθηση μετώπου](#)

Αν θέλαμε να παρουσιάσουμε πιο αναλυτικά τα βήματα του γενικού αλγόριθμου, παρακολουθώντας τη διαμόρφωση του μετώπου ταυτόχρονα με τη διαμόρφωση της ουράς και χρησιμοποιώντας τον όρο κατάσταση αντί κόμβου, θα είχαμε την ακόλουθη περιγραφή βημάτων:

1. Βάλε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης. Βάλε το αρχικό μονοπάτι που περιέχει την αρχική κατάσταση μέσα στην κενή ουρά.
2. Αν το μέτωπο είναι κενό, τότε η αναζήτηση ολοκληρώθηκε ανεπιτυχώς, χωρίς να βρεθεί λύση, ΤΕΛΟΣ.
3. Αν η πρώτη σε σειρά κατάσταση του μετώπου είναι στόχος, τότε επέστρεψε το πρώτο μονοπάτι της ουράς ως λύση, ΤΕΛΟΣ.
4. Αν η πρώτη σε σειρά κατάσταση του μετώπου ανήκει στο κλειστό σύνολο, τότε Πήγαινε στο βήμα 2.
5. Εφάρμοσε τους τελεστές μετάβασης στην πρώτη στη σειρά κατάσταση του μετώπου, για να γίνει κατάσταση-γονέας και να παραχθούν οι διάδοχες καταστάσεις. Δημιούργησε τα αντίστοιχα επεκτεταμένα μονοπάτια που καταλήγουν σε αυτές.
6. Πρόσθεσε τις διάδοχες καταστάσεις στο μέτωπο, αφού αφαιρέσεις την κατάσταση-γονέα, και δημιούργησε το νέο μέτωπο. Πρόσθεσε τα επεκτεταμένα μονοπάτια στην ουρά, αφού αφαιρέσεις το μονοπάτι που οδηγούσε στον κόμβο-γονέα, και δημιούργησε τη νέα ουρά.
7. Αν το απαιτεί ο αλγόριθμος αναζήτησης, αναδιάταξε τις καταστάσεις του μετώπου σύμφωνα με το ποιοτικό κριτήριο που ορίζει η στρατηγική του αλγόριθμου και πραγματοποίησε την αντίστοιχη αναδιάταξη στα μονοπάτια της ουράς.
8. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
9. Πήγαινε στο βήμα 2.

Η δεύτερη εκδοχή παρουσίασης των βημάτων της διαδικασίας αναζήτησης με παρακολούθηση του μετώπου και ταυτόχρονα της ουράς βασίζεται στην περιγραφή της επέκτασης των μερικών λύσεων που αντιπροσωπεύει η ουρά μονοπατιών.

Δείτε [ρομπ 1.2 - Ψευδοκώδικας Γενικού Αλγόριθμου Αναζήτησης με παρακολούθηση μετώπου και ουράς](#)

Οι μέθοδοι αναζήτησης χωρίζονται σε δύο μεγάλες κατηγορίες: στις τυφλές μεθόδους και τις ευρετικές. Οι αλγόριθμοι αναζήτησης των τυφλών μεθόδων τοποθετούν τα μονοπάτια στην ουρά βάσει του χρόνου δημιουργίας τους. Αντίθετα, στους ευρετικούς αλγόριθμους τα μονοπάτια της ουράς αναδιατάσσονται βάσει των ποιοτικών κριτηρίων που τα χαρακτηρίζουν.

Με απλές προσαρμογές, ο γενικός αλγόριθμος αναζήτησης μετατρέπεται εύκολα σε συγκεκριμένο τυφλό ή ευρετικό αλγόριθμο.

1.4 Τυφλές μέθοδοι αναζήτησης

Οι **τυφλές μέθοδοι αναζήτησης** (blind search methods) εφαρμόζονται σε προβλήματα στα οποία είτε δεν μας ενδιαφέρει να βρεθεί μια ποιοτική λύση είτε δε διαθέτουν πληροφορίες που να επιτρέπουν στον αλγόριθμο αναζήτησης την επιλογή του ποιοτικά καλύτερου κόμβου μέσα από το μέτωπο αναζήτησης, όταν αυτός επιχειρεί να αναπτύξει το δένδρο αναζήτησης.

Με τις τυφλές μεθόδους αναζήτησης, αν υπάρχει λύση, τότε η μέθοδος εγγυάται την εύρεσή της. Η διαφορά τους έγκειται στον τρόπο (τη χρονική σειρά) με τον οποίο αναπτύσσουν τα μονοπάτια μέσα στο δένδρο αναζήτησης.

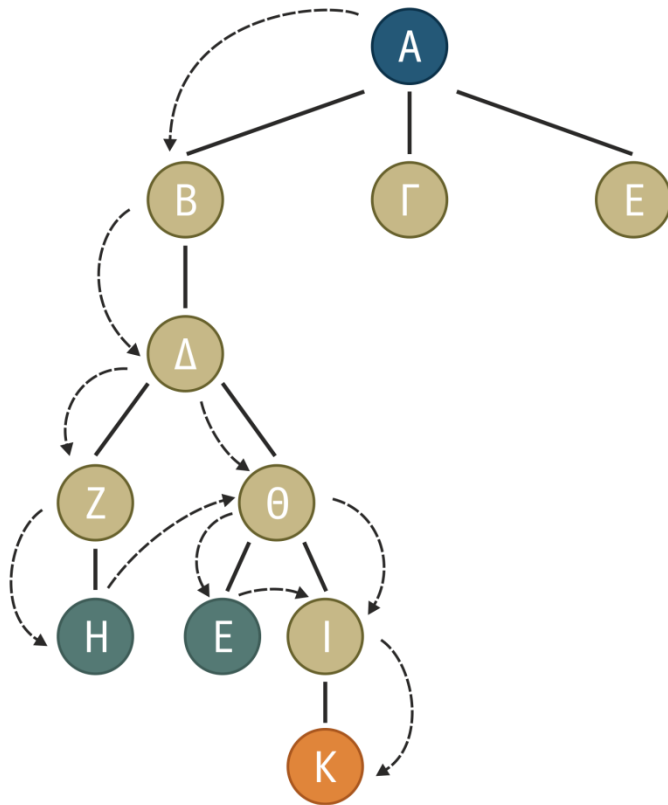
Γνωστές τυφλές μέθοδοι αναζήτησης είναι οι:

- Πρώτα σε Βάθος Αναζήτηση,
- Πρώτα σε Πλάτος Αναζήτηση,
- Αναζήτηση Επαναληπτικής Εμβάθυνσης,
- Αναζήτηση Διπλής Κατεύθυνσης.

1.4.1. Πρώτα σε Βάθος Αναζήτηση

Δεδομένου ότι σε μια τυφλή μέθοδο αναζήτησης ένα μονοπάτι είναι εξίσου καλό με οποιοδήποτε άλλο, η **Πρώτα σε Βάθος Αναζήτηση** (Depth-First Search - DFS) επεκτείνει κάθε φορά το αριστερότερο από τα υπάρχοντα μονοπάτια του δένδρου, αρχίζοντας από το μονοπάτι που περιέχει τη ρίζα του δένδρου. Ουσιαστικά, επεκτείνει το μέτωπο αναζήτησης προς αριστερά και προς το βάθος του δένδρου αναζήτησης. Η διαδικασία ολοκληρώνεται, όταν εντοπιστεί ένας κόμβος που αντιστοιχεί σε μια επιθυμητή τελική κατάσταση ή όταν εξαντληθεί η αναζήτηση, δηλαδή όταν όλα τα μονοπάτια καταλήξουν σε κόμβους που δεν μπορούν να επιλεγούν ως γονικοί κόμβοι, γιατί οδηγούν σε κύκλους.

Στο σχήμα 1.18 παρουσιάζεται η πορεία που ακολουθεί η Πρώτα σε Βάθος Αναζήτηση, καθώς σχηματίζει βήμα προς βήμα το δένδρο αναζήτησης του προβλήματος του περιοδεύοντος πωλητή, στην προσπάθειά της να οδηγήσει τον πωλητή από την πόλη Α στην πόλη Κ. Η λύση που βρίσκει ο αλγόριθμος είναι το μονοπάτι Α Β Δ Θ Ι Κ.



Σχήμα 1.18 Η πορεία σχηματισμού του δένδρου αναζήτησης που ακολουθεί η DFS κατά την επίλυση του προβλήματος του περιοδεύοντος πωλητή

Δείτε [κινούμενη εικόνα 1.2 - Ανάπτυξη Δένδρου Αναζήτησης με DFS και παράλληλη παρακολούθηση μετώπου](#)

Τα βήματα της Πρώτα σε Βάθος Αναζήτησης με παρακολούθηση μετώπου είναι τα ακόλουθα:

1. Βάλτε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης.
2. Αν το μέτωπο είναι κενό, τότε η αναζήτηση ολοκληρώθηκε ανεπιτυχώς, χωρίς να βρεθεί λύση, ΤΕΛΟΣ.
3. Αν ο πρώτος σε σειρά κόμβος του μετώπου είναι στόχος, τότε βρέθηκε λύση, ΤΕΛΟΣ.
4. Αν ο πρώτος σε σειρά κόμβος του μετώπου ανήκει στο κλειστό σύνολο, τότε Πήγαινε στο βήμα 2.
5. Εφάρμοσε τους τελεστές στον πρώτο σε σειρά κόμβο του μετώπου, για να γίνει γονικός κόμβος και να παραχθούν οι διάδοχοι κόμβοι.
6. Πρόσθεσε τους διάδοχους κόμβους στην αρχή του μετώπου, αφού αφαιρέσεις το γονικό κόμβο, και δημιούργησε το νέο μέτωπο.
7. Βάλτε το γονικό κόμβο στο κλειστό σύνολο.
8. Πήγαινε στο βήμα 2.

Στον παρακάτω πίνακα 1.3 παρουσιάζονται οι πρώτοι έξι κύκλοι ανάπτυξης του δένδρου αναζήτησης της DFS για το πρόβλημα του περιοδεύοντος πωλητή με παρακολούθηση του μετώπου αναζήτησης.

Μέτωπο Αναζήτησης	Γονικός Κόμβος	Διάδοχοι Κόμβοι	Κλειστό Σύνολο
<A>	A	B, Γ, E	{}
<B Γ E>	B	Δ	{A}
<Δ Γ E>	Δ	Z, θ	{A, B}
<Z θ Γ E>	Z	H	{A, B, Δ}
<H θ Γ E>	H	-	{A, B, Δ, Z}
<θ Γ E>	θ	E, I	{A, B, Δ, Z, H}

Πίνακας 1.3 Οι έξι πρώτοι κύκλοι λειτουργίας της DFS για το πρόβλημα του περιοδεύοντος πωλητή με μέτωπο αναζήτησης

Η διαφοροποίηση της λειτουργίας της DFS από άλλες τυφλές μεθόδους αναζήτησης βρίσκεται στο βήμα 7, όπου προσδιορίζεται η θέση του μετώπου στην οποία θα τοποθετηθούν οι κόμβοι-παιδιά. Η εξέταση αμέσως προηγούμενων (χρονικά) καταστάσεων, όπως υλοποιείται στο βήμα 4, ονομάζεται **χρονική οπισθοδρόμηση** (chronological backtracking).

Ακολουθεί ο ψευδοκώδικας του αλγόριθμου της DFS με παρακολούθηση μετώπου και ουράς.

ΑΡΧΗ

Βάλε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης. **Βάλε το αρχικό μονοπάτι που περιέχει την αρχική κατάσταση μέσα στην κενή ουρά.**

Έως ότου η πρώτη κατάσταση του μετώπου ή το μέτωπο αδειάσει,

Εφαρμοσε τους τελεστές μετάβασης στην πρώτη στη σειρά κατάσταση του μετώπου, για να γίνει κατάσταση-γονέας και να παραχθούν οι διάδοχες καταστάσεις. **Δημιούργησε τα αντίστοιχα επεκτεταμένα μονοπάτια που καταλήγουν σε αυτές.**

Εάν υπάρχουν διάδοχες καταστάσεις,

Πρόσθεσε τις διάδοχες καταστάσεις στην ΑΡΧΗ του μετώπου, αφού αφαιρέσεις την κατάσταση-γονέα, και δημιούργησε το νέο μέτωπο. **Πρόσθεσε τα επεκτεταμένα μονοπάτια στην ΑΡΧΗ της ουράς,** αφού αφαιρέσεις το μονοπάτι που οδηγούσε στην κατάσταση-γονέα, και δημιούργησε τη νέα ουρά.

Τέλος έως ότου

Εάν το μέτωπο δεν είναι άδειο,

Παράδωσε το πρώτο μονοπάτι της ουράς ως λύση.

ΤΕΛΟΣ

Δείτε [κινούμενη εικόνα 1.3 - Ανάπτυξη Δένδρου Αναζήτησης με DFS και παράλληλη παρακολούθηση ουράς](#)

Στον παρακάτω πίνακα 1.4 παρουσιάζονται οι πρώτοι έξι κύκλοι ανάπτυξης του δένδρου αναζήτησης της DFS για το πρόβλημα του περιοδεύοντος πωλητή με παρακολούθηση ουράς αναζήτησης.

Ουρά Αναζήτησης	Μονοπάτι για Επέκταση	Διάδοχοι Κόμβοι	Διάδοχα Μονοπάτια	Κλειστό Σύνολο
<(A)>	(A)	B, Γ, E	(A B), (A Γ), (A E)	{}
<(A B) (A Γ) (A E)>	(A B)	Δ	(A B Δ)	{A}
<(A B Δ) (A Γ) (A E)>	(A B Δ)	Z, Θ	(A B Δ Z), (A B Δ Θ)	{A, B}
<(A B Δ Z) (A B Δ Θ) (A Γ) (A E)>	(A B Δ Z)	H	(A B Δ Z H)	{A, B, Δ}
<(A B Δ Z H) (A B Δ Θ) (A Γ) (A E)>	(A B Δ Z H)	-		{A, B, Δ, Z}
<(A B Δ Θ) (A Γ) (A E)>	(A B Δ Θ)	E, I	(A B Δ Θ E), (A B Δ Θ I)	{A, B, Δ, Z, H}

Πίνακας 1.4 Οι έξι πρώτοι κύκλοι λειτουργίας της DFS για το πρόβλημα του περιοδεύοντος πωλητή με ουρά αναζήτησης

Πλεονεκτήματα της DFS

- Το μέτωπο της αναζήτησης δε μεγαλώνει πάρα πολύ.
- Η DFS εγγυάται πάντα την εύρεση λύσης, αν αυτή υπάρχει, με την προϋπόθεση ότι ο χώρος αναζήτησης είναι πεπερασμένος.

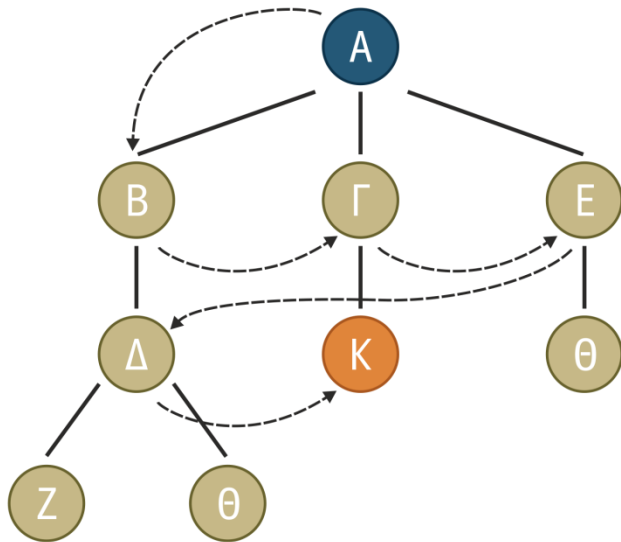
Μειονεκτήματα της DFS

- Δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη.
- Θεωρείται μη πλήρης σε μη πεπερασμένους χώρους αναζήτησης.
- Η DFS δεν εγγυάται την εύρεση λύσης, αν ο χώρος αναζήτησης δεν είναι πεπερασμένος.

1.4.2 Η Πρώτα σε Πλάτος Αναζήτηση

Η **Πρώτα σε Πλάτος Αναζήτηση** (Breadth-First Search - BFS) εξετάζει πρώτα τα μονοπάτια που βρίσκονται στο ίδιο βάθος του δένδρου αναζήτησης. Μόνο όταν τα εξετάσει όλα, αρχίζει την επέκτασή τους στο αμέσως επόμενο επίπεδο. Ουσιαστικά, επεκτείνει το μέτωπο αναζήτησης ανά επίπεδο του δένδρου αναζήτησης. Η διαδικασία ολοκληρώνεται, όταν εντοπιστεί ένας κόμβος που αντιστοιχεί σε μια επιθυμητή τελική κατάσταση ή όταν εξαντληθεί η αναζήτηση, δηλαδή όταν όλα τα μονοπάτια καταλήξουν σε κόμβους που δεν μπορούν να επιλεγούν ως γονικοί κόμβοι, γιατί οδηγούν σε κύκλους.

Στο σχήμα 1.19 παρουσιάζεται η πορεία που ακολουθεί η Πρώτα σε Πλάτος Αναζήτηση, καθώς σχηματίζει βήμα-βήμα το δένδρο αναζήτησης στην προσπάθειά της να επιλύσει το πρόβλημα του περιοδεύοντος πωλητή, δηλαδή να οδηγήσει τον πωλητή από την πόλη A στην πόλη K. Η λύση που θα βρεθεί είναι το μονοπάτι A Γ K.



Σχήμα 1.19 Η πορεία σχηματισμού του δένδρου αναζήτησης που ακολουθεί η Πρώτα σε Πλάτος Αναζήτηση κατά την επίλυση του προβλήματος του περιοδεύοντος πωλητή

Τα βήματα της BFS με παρακολούθηση μετώπου είναι τα ακόλουθα:

1. Βάλτε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης.
2. Αν το μέτωπο είναι κενό, τότε η αναζήτηση ολοκληρώθηκε ανεπιτυχώς, χωρίς να βρεθεί λύση, ΤΕΛΟΣ.
3. Αν ο πρώτος σε σειρά κόμβος του μετώπου είναι στόχος, τότε βρέθηκε λύση, ΤΕΛΟΣ.
4. Αν ο πρώτος σε σειρά κόμβος του μετώπου ανήκει στο κλειστό σύνολο, τότε Πήγαινε στο βήμα 2.
5. Εφάρμοσε τους τελεστές στον πρώτο σε σειρά κόμβο του μετώπου, για να γίνει γονικός κόμβος και να παραχθούν οι διάδοχοι κόμβοι.
6. Πρόσθεσε τους διάδοχους κόμβους **στο τέλος του μετώπου**, αφού αφαιρέσεις το γονικό κόμβο, και δημιούργησε το νέο μέτωπο.
7. Βάλτε το γονικό κόμβο στο κλειστό σύνολο.
8. Πήγαινε στο βήμα 2.

Δείτε [κινούμενη εικόνα 1.4 - Ανάπτυξη Δένδρου Αναζήτησης με BFS και παράλληλη παρακολούθηση μετώπου](#)

Στον παρακάτω πίνακα 1.5 παρουσιάζεται η ανάπτυξη του δένδρου αναζήτησης της BFS με παρακολούθηση μετώπου αναζήτησης για το πρόβλημα του περιοδεύοντος πωλητή μέχρι την εύρεση στόχου.

Μέτωπο Αναζήτησης	Γονικός Κόμβος	Διάδοχοι Κόμβοι	Κλειστό Σύνολο
<A>	A	B, Γ, E	{}
<B Γ E>	B	Δ	{A}
<Γ E Δ>	Γ	Κ	{A, B}
<E Δ Κ>	E	Θ	{A, B, Γ}
<Δ Κ Θ>	Δ	Z, Θ	{A, B, Γ, E}
<Κ Θ Z Θ>	Κ	ΒΡΕΘΗΚΕ ΣΤΟΧΟΣ	

Πίνακας 1.5 Οι κύκλοι λειτουργίας της BFS για το πρόβλημα του περιοδεύοντος πωλητή με μέτωπο αναζήτησης

Ακολουθεί ο ψευδοκώδικας του αλγόριθμου της BFS με παρακολούθηση μετώπου και ουράς.

ΑΡΧΗ

Βάλτε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης. **Βάλτε το αρχικό μονοπάτι που περιέχει την αρχική κατάσταση μέσα στην κενή ουρά.**

Έως ότου η πρώτη κατάσταση του μετώπου ή το μέτωπο αδειάσει,

Εφαρμόστε τους τελεστές μετάβασης στην πρώτη στη σειρά κατάσταση του μετώπου, για να γίνει κατάσταση-γονέας και να παραχθούν οι διάδοχες καταστάσεις. Δημιούργησε τα αντίστοιχα επεκτεταμένα μονοπάτια που καταλήγουν σε αυτές.

Εάν υπάρχουν διάδοχες καταστάσεις, πρόσθεσε τις διάδοχες καταστάσεις στο **ΤΕΛΟΣ του μετώπου**, αφού αφαιρέσεις την κατάσταση-γονέα, και δημιούργησε το νέο μέτωπο. Πρόσθεσε τα επεκτεταμένα μονοπάτια στο **ΤΕΛΟΣ της ουράς**, αφού αφαιρέσεις το μονοπάτι που οδηγούσε στην κατάσταση-γονέα, και δημιούργησε τη νέα ουρά.

Τέλος έως ότου

Εάν το μέτωπο δεν είναι άδειο,

Παράδωσε το πρώτο μονοπάτι της ουράς ως λύση.

ΤΕΛΟΣ

Δείτε [κινούμενη εικόνα 1.5 - Ανάπτυξη Δένδρου Αναζήτησης με BFS και παράλληλη παρακολούθηση ουράς](#)

Στον παρακάτω πίνακα 1.6 παρουσιάζεται η ανάπτυξη του δένδρου αναζήτησης της BFS με παρακολούθηση ουράς για το πρόβλημα του περιοδεύοντος πωλητή μέχρι την εύρεση λύσης.

Ουρά Αναζήτησης	Μονοπάτι για Επέκταση	Διάδοχοι Κόμβοι	Διάδοχα Μονοπάτια	Κλειστό Σύνολο
<(A)>	(A)	B, Γ, E	(A B), (A Γ), (A E)	{}
<(A B) (A Γ) (A E)>	(A B)	Δ	(A B Δ)	{A}
<(A Γ) (A E) (A B Δ)>	(A Γ)	Κ	(A Γ Κ)	{A, B}
<(A E) (A B Δ) (A Γ Κ)>	(A E)	Θ	(A E Θ)	{A, B, Γ}
<(A B Δ) (A Γ Κ) (A E Θ)>	(A B Δ)	Z, Θ	(A B Δ Z), (A B Δ Θ)	{A, B, Γ, Θ}
<(A Γ Κ) (A E Θ) (A B Δ Z) (A B Δ Θ)>	(A Γ Κ)	ΒΡΕΘΗΚΕ ΛΥΣΗ		

Πίνακας 1.6 Οι κύκλοι λειτουργίας της BFS για το πρόβλημα του περιοδεύοντος πωλητή με ουρά αναζήτησης

Πλεονεκτήματα της BFS

- Και εδώ η αναζήτηση εγγυάται πάντα την εύρεση λύσης, αν αυτή υπάρχει, με την προϋπόθεση ότι ο χώρος αναζήτησης είναι πεπερασμένος.
- Βρίσκει πάντα τη συντομότερη λύση (μικρότερη σε μήκος μονοπατιού).
- Είναι πλήρης.

Μειονεκτήματα της BFS

- Το μέτωπο αναζήτησης μπορεί να γίνεται προοδευτικά πολύ μεγάλο. Αντίστοιχα, οι απαιτήσεις σε μνήμη για την αποθήκευση των μονοπατιών της ουράς μεγαλώνει εκθετικά, όσο βαθαίνει το μέτωπο αναζήτησης.

Το να κρίνουμε ποια από τις 2 τυφλές μεθόδους, DFS και BFS, είναι προσφορότερη εξαρτάται άμεσα από το πρόβλημα προς επίλυση. Γενικά, πρέπει να αποφεύγεται η BFS, όταν είναι γνωστό ότι οι υπάρχουσες λύσεις βρίσκονται σε μεγάλο βάθος. Αντίστοιχα, η χρήση της DFS είναι προτιμότερη, όταν ο στόχος βρίσκεται σε μονοπάτια μεγάλου μήκους.

1.4.3 Αλγόριθμος Αναζήτησης Επαναληπτικής Εμβάθυνσης

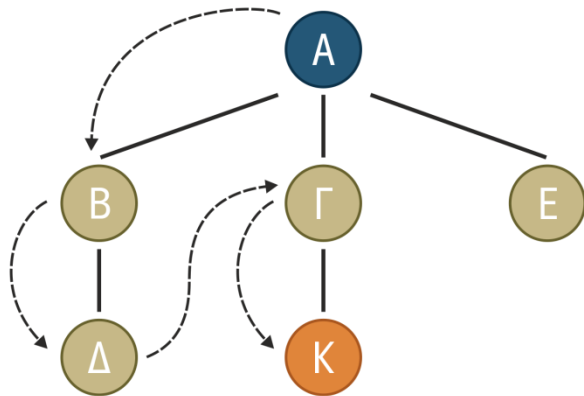
Η μέθοδος **Αναζήτησης Επαναληπτικής Εμβάθυνσης** (Iterative Deepening Search - IDS) αναζητά τη λύση με τη λογική της DFS, αλλά περιορίζει το βάθος της αναζήτησης θέτοντας ένα μικρό βάθος στην αρχή ως όριο, το οποίο σταδιακά αυξάνεται με εμβάθυνση. Αυτό αποτελεί έναν τρόπο αντιμετώπισης του προβλήματος εύρεσης σωστού ορίου για το βάθος αναζήτησης. Συνήθως, η μέθοδος δοκιμάζει όλα τα πιθανά βάθη, αυξάνοντας προοδευτικά το βάθος κατά 1 αρχίζοντας από το βάθος 0, π.χ. 0, 1, 2, κτλ.

Βήματα αλγόριθμου Επαναληπτικής εκβάθυνσης:

1. Όρισε το αρχικό βάθος αναζήτησης (συνήθως 0).
2. Εφάρμοσε τον αλγόριθμο DFS μέχρι αυτό το βάθος αναζήτησης.
3. Αν έχεις βρει λύση, σταμάτησε.
4. Αύξησε το βάθος αναζήτησης (συνήθως κατά 1).
5. Πήγαινε στο βήμα 2.

Στο παράδειγμα του σχήματος 1.20, ο αλγόριθμος επαναληπτικής εμβάθυνσης θα δημιουργήσει το δένδρο αναζήτησης σε 3 κύκλους.

-
- 1ος κύκλος: το βάθος αναζήτησης είναι 0 και δημιουργείται μόνο ο κόμβος-ρίζα A.
- 2ος κύκλος: σε βάθος αναζήτησης με βάθος 1, οπότε δημιουργούνται και ελέγχονται οι κόμβοι A,B,Γ,E.
- 3ος και τελικός κύκλος: το βάθος αναζήτησης είναι 2 και δημιουργούνται οι κόμβοι A, B, Γ, Δ E και στη συνέχεια ο K, που είναι ο στόχος. Έτσι η αναζήτηση ολοκληρώνεται.



Σχήμα 1.20 Η πορεία σχηματισμού δένδρου αναζήτησης που ακολουθεί η επαναληπτική εμβάθυνση σε βάθος αναζήτησης $n=2$

Δείτε [κινούμενη εικόνα 1.6 - Υποδειγματική Ανάπτυξη Δένδρου Αναζήτησης με την IDS για το πρόβλημα των 8 πλακιδίων μέχρι βάθους \$n=2\$](#)

Πλεονεκτήματα της IDS

- Είναι πλήρης.
- Αν το βάθος αυξάνεται κατά 1 σε κάθε κύκλο και ο αλγόριθμος IDS βρει λύση, τότε αυτή η λύση θα είναι η καλύτερη χρονικά.
- Συνδυάζει τα πλεονεκτήματα των DFS και BFS και έχει αποδειχθεί ότι έχει την ίδια πολυπλοκότητα σε χώρο και χρόνο με αυτούς.
- Δεν κινδυνεύει να χαθεί σε κάποιο κλαδί απείρου μήκους.

Μειονεκτήματα της IDS:

- Δεν θυμάται τίποτα από κάθε προηγούμενο κύκλο αναζήτησης.

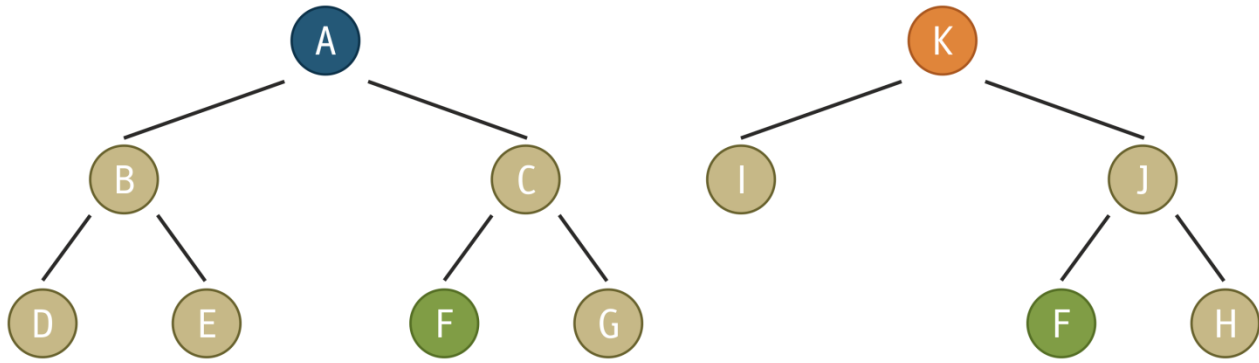
1.4.4 Μέθοδος Αναζήτησης Διπλής Κατεύθυνσης

Η μέθοδος **Αναζήτησης Διπλής Κατεύθυνσης** (Bidirectional Search) πραγματοποιεί αναζήτηση ταυτόχρονα από την αρχική κατάσταση προς το στόχο και από το στόχο προς την αρχική κατάσταση. Για να εφαρμοστεί, θα πρέπει οι ενέργειες που πραγματοποιούν οι τελεστές μετάβασης του προβλήματος να είναι *αναστρέψιμες* (*reversible*) και να είναι πλήρως γνωστός ο στόχος του προβλήματος. Η αναζήτηση σταματά, όταν οι δύο επιμέρους αναζητήσεις συναντηθούν.

Βασικά βήματα Αλγόριθμου Διπλής Κατεύθυνσης

Αρχισε την αναζήτηση από την αρχική και τελική κατάσταση ταυτόχρονα.
Αν κάποια κατάσταση που επεκτείνεται είναι κοινή, τότε βρέθηκε λύση.

Λύση στην περίπτωση της αναζήτησης του σχήματος 1.21 είναι η ένωση του μονοπατιού που ξεκινά από τον κόμβο A και καταλήγει στον κοινό κόμβο των δύο αναζητήσεων, δηλαδή το F, και του μονοπατιού που ξεκινά από τον κόμβο-στόχο K και καταλήγει επίσης στο F, δηλαδή λύση είναι το μονοπάτι A, C, F, J, K.



Σχήμα 1.21 Παράδειγμα δένδρων αναζήτησης διπλής κατεύθυνσης

Παρά το ότι η εφαρμογή του αλγόριθμου υπόσχεται σημαντική μείωση του χρόνου αναζήτησης, ένα σύνολο από προβλήματα μπορούν να προκύψουν, όπως τα ακόλουθα:

- Τι σημαίνει αναζήτηση από το στόχο προς τα πίσω στην αρχική κατάσταση;
- Τι γίνεται, όταν έχουμε πολλές πιθανές καταστάσεις στόχου;
- Υπάρχει κίνδυνος να μη συναντηθούν οι δυο αναζητήσεις και σε ποια περίπτωση μπορεί αυτό να συμβεί;
- Μπορούμε να ελέγξουμε αποδοτικά πότε συναντώνται οι δύο αναζητήσεις;
- Τι είδους μέθοδο αναζήτησης επιλέγουμε σε καθεμία αναζήτηση;

Η εφαρμογή ταυτόχρονα της DFS και της BFS αναζήτησης απαντά σε μερικά από τα παραπάνω ερωτήματα και εγγυάται ότι σίγουρα οι δυο αναζητήσεις θα συναντηθούν, αλλά παραμένει πρόβλημα σε ποια κατεύθυνση να εφαρμοστεί η καθεμία από αυτές, ώστε να συναντηθούν πιο αποδοτικά.

Μειονέκτημα της αναζήτησης διπλής κατεύθυνσης είναι ότι υπάρχει επιπλέον κόστος επικοινωνίας μεταξύ των δύο αναζητήσεων.

1.5 Ευρετικές μέθοδοι αναζήτησης

Η εκθετική πολυπλοκότητα του χρόνου που απαιτούν οι μέθοδοι τυφλής αναζήτησης είναι ασύμφορη για την επιλογή τους, όταν τα προβλήματα προς επίλυση είναι προβλήματα του πραγματικού κόσμου. Η πρόταση στην περίπτωση αυτή είναι η χρήση ενός ευρετικού μηχανισμού με τέτοιο τρόπο, ώστε η επιλογή κόμβων για επίσκεψη να γίνεται σύμφωνα με κάποιο ποιοτικό κριτήριο και όχι τυφλά. Ο **ευρετικός μηχανισμός** (heuristic) συνήθως χρησιμοποιεί την κοινή λογική, για να εκφράσει το κριτήριο που θα αποτελέσει κατευθυντήρια ιδέα για τη χάραξη πορείας κατά την αναζήτηση. Η κατευθυντήρια ιδέα στοχεύει στο να επιλέγεται κάθε φορά η περισσότερα υποσχόμενη πορεία. Για παράδειγμα, στο πρόβλημα του περιοδεύοντος πωλητή, κατευθυντήρια ιδέα θα μπορούσε να είναι ότι «προσφορότερη πόλη, για να γίνει γονικός κόμβος, είναι εκείνη της οποίας η απόσταση από την πόλη-στόχο σε ευθεία γραμμή είναι η μικρότερη».

Η **ευρετική αναζήτηση** (heuristic search) είναι η μέθοδος αναζήτησης που υποστηρίζεται από κάποιον ευρετικό μηχανισμό. Βασίζεται στη γνώση για το συγκεκριμένο πρόβλημα προς επίλυση και στην κοινή λογική· συνήθως χρησιμοποιείται για εύρεση μίας «καλής» λύσης και όχι απαραίτητα της άριστης. Στην πράξη, σχηματίζει το δένδρο αναζήτησης αναδιατάσσοντας σε κάθε κύκλο λειτουργίας το μέτωπο αναζήτησης σύμφωνα με το κριτήριο που χρησιμοποιεί, χωρίς να ερευνά αν είναι ορθό ή λανθασμένο. Το κριτήριο προκύπτει μέσω μιας ευρετικής συνάρτησης. Μια ευρετική αναζήτηση δεν εγγυάται πάντα την εύρεση λύσης, αν αυτή υπάρχει.

Γνωστές ευρετικές μέθοδοι αναζήτησης είναι οι παρακάτω καθώς και πολλές ακόμα που αποτελούν παραλλαγές ή και συνδυασμούς τους:

- Καλύτερη Πρώτη – Best-First,

- Αναρρίχηση Λόφων – Hill-Climbing,
- Ακτινωτή Αναζήτηση – Beam Search,
- Αναζήτηση Βασικής Επέκτασης και Οριοθέτησης – Basic Branch & Bound Search,
- Άλφα-Άστρο - A*.

Επίσης, γνωστές ευρετικές μέθοδοι αναζήτησης για προβλήματα παιχνιδιών 2 ατόμων είναι οι:

- Αναζήτηση Μέγιστου-Ελάχιστου,
- Άλφα-Βήτα.

Στο εγχειρίδιο αυτό δεν θα ασχοληθούμε με αλγόριθμους επίλυσης παιχνιδιών δύο ατόμων ή άλλων παρόμοιων προβλημάτων φυσικού κόσμου.

1.5.1 Ευρετικές Συναρτήσεις

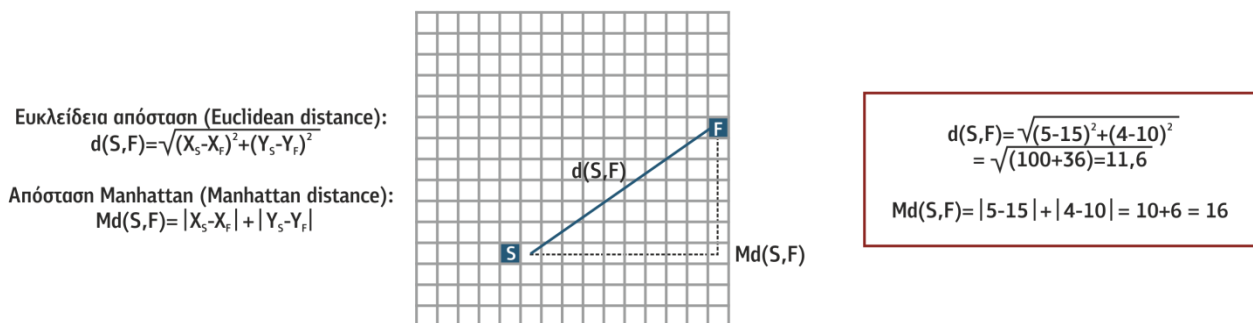
Ευρετική συνάρτηση (heuristic function) είναι η υλοποίηση ενός ευρετικού μηχανισμού. Έχει πεδίο ορισμού το χώρο καταστάσεων ενός προβλήματος και πεδίο τιμών το σύνολο των τιμών που αντιστοιχούν σε αυτές. Επιστρέφει ευρετικές τιμές που προκύπτουν ως αποτέλεσμα υπολογισμών βασισμένων στη γνώση που διατίθεται για το πρόβλημα και την κατάσταση που εξετάζεται.

Ευρετική τιμή (heuristic value) είναι η τιμή της ευρετικής συνάρτησης και εκφράζει το κόστος αναζήτησης. Μπορεί σε μια απλή προσέγγιση να εκφράζει απλώς το κόστος μετάβασης από τον τρέχοντα κόμβο σε έναν κόμβο-στόχο, αλλά για εύρεση άριστων λύσεων πρέπει να εκφράζει μια συν-εκτίμηση του κόστους της διαδρομής έως τον τρέχοντα κόμβο και του εκτιμώμενου κόστους από το σημείο αυτό έως τον κόμβο-στόχο.

Η επιλογή μιας ευρετικής συνάρτησης πρέπει να βασίζεται στο:

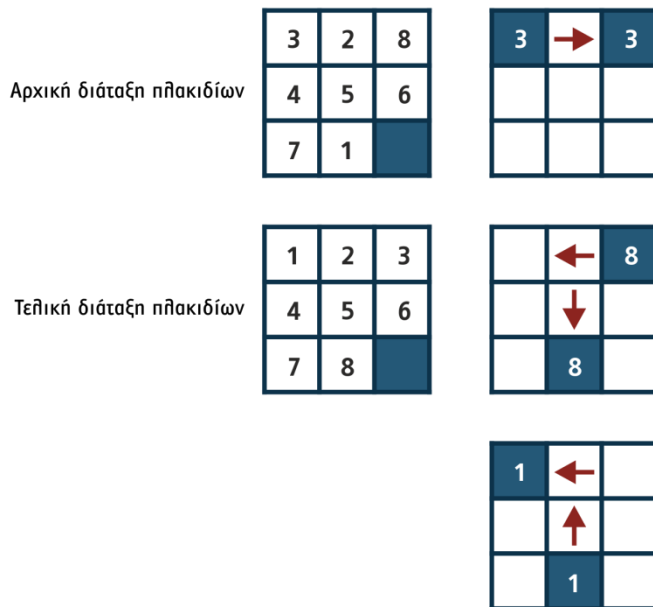
- πόσο ακριβής είναι,
- πόσα μεταεπίπεδα απαιτεί, δηλαδή σε ποιο βάθος θα φτάσει η αναζήτηση,
- αν είναι ικανή να βρει την καλύτερη λύση.

Γνωστοί ευρετικοί μηχανισμοί είναι η Ευκλείδεια απόσταση και η απόσταση Manhattan που υπολογίζονται, όπως φαίνεται στο παρακάτω σχήμα 1.22.



Σχήμα 1.22 Υπολογισμός Ευκλείδειας απόστασης και απόστασης Manhattan

Για παράδειγμα, στο πρόβλημα των 8 πλακιδίων μπορούμε να θεωρήσουμε ως ευρετικό μηχανισμό τη διαφορά των θέσεων των πλακιδίων της τρέχουσας κατάστασης από τις αντίστοιχες θέσεις των ίδιων πλακιδίων στην τελική θέση και ευρετική συνάρτηση το άθροισμα των αποστάσεων Manhattan κάθε πλακιδίου από την τελική του θέση.



Σχήμα 1.23 Αποστάσεις Manhattan για το πρόβλημα των 8 πλακιδίων

Στην περίπτωση του σχήματος 1.23, στην αρχική κατάσταση μόνο τα πλακίδια “3”, “8” και “1” είναι τοποθετημένα με λανθασμένο τρόπο σε σχέση με τον επιδιωκόμενο στόχο. Η απόσταση Manhattan αυτών από την τελική τους θέση είναι 2, 3, και 3 αντίστοιχα, ενώ των υπολοίπων η αντίστοιχη απόσταση είναι 0. Έτσι, αν ως S_i χαρακτηρίσουμε ένα πλακίδιο της κατάστασης S που είναι σε λάθος θέση, η ευρετική συνάρτηση θα επιστρέψει 8, που είναι το άθροισμα των επιμέρους αποστάσεων Manhattan.

Manhattan Distance κατάστασης S :

$$h(S) = \sum_i^3 |XSi - XFi| + |YSi - YFi| = 2 + 3 + 3 = 8$$

Ένα εναλλακτικό ευρετικό κριτήριο για το πρόβλημα των 8 πλακιδίων είναι ο υπολογισμός του πλήθους των πλακιδίων που είναι σε λάθος θέση. Για την κατάσταση S του σχήματος 1.23, επειδή υπάρχουν μόνο 3 πλακίδια σε λάθος θέση, τα “3”, “8” και “1”, η ευρετική συνάρτηση θα επιστρέψει 3: $h(S) = 3$.

1.5.2 Αναζήτηση Καλύτερης-Πρώτης

Η μέθοδος αναζήτησης **Πρώτα στο Καλύτερο** ή **Καλύτερη-Πρώτη** (Best First Search - BestFS) σε κάθε κύκλο λειτουργίας της επισκέπτεται την κατάσταση του μετώπου αναζήτησης την οποία το κριτήριο θεωρεί «καλύτερη» βάσει ενός ευρετικού κριτηρίου άμεσα εξαρτώμενου από το πρόβλημα προς επίλυση. Χαρακτηριστικό της μεθόδου είναι ότι σε κάθε βήμα αναζήτησης η μέθοδος επιλέγει προς επέκταση το πιο πολλά υποσχόμενο μονοπάτι μεταξύ των μονοπατιών που βρίσκονται στην ουρά.

Ψευδοκώδικας αλγόριθμου BestFS

ΑΡΧΗ

Βάλε την αρχική κατάσταση στο κενό μέτωπο αναζήτησης. Βάλε το αρχικό μονοπάτι που περιέχει την αρχική κατάσταση μέσα στην κενή ουρά.

Έως ότου η πρώτη κατάσταση του μετώπου ή το μέτωπο αδειάσει,

Εφαρμόσε τους τελεστές μετάβασης στην πρώτη στην σειρά κατάσταση του μετώπου, για να γίνει κατάσταση-γονέα και να παραχθούν οι διάδοχες καταστάσεις. Δημιούργησε τα αντίστοιχα επεκτεταμένα μονοπάτια που καταλήγουν σε αυτές.

Εάν υπάρχουν διάδοχες καταστάσεις,

πρόσθεσε τις διάδοχες καταστάσεις στο μετώπο, αφού αφαιρέσεις την κατάσταση-γονέα. Πρόσθεσε τα επεκτεταμένα μονοπάτια στην ουρά, αφού αφαιρέσεις το μονοπάτι που οδηγούσε στην κατάσταση-γονέα.

Αναδιάταξε τις καταστάσεις μέσα στο μέτωπο βάσει ενός ευρετικού κριτηρίου δημιουργώντας το νέο μέτωπο και αναδιάταξε ανάλογα τα μονοπάτια της ουράς δημιουργώντας τη νέα ουρά.

Τέλος έως ότου

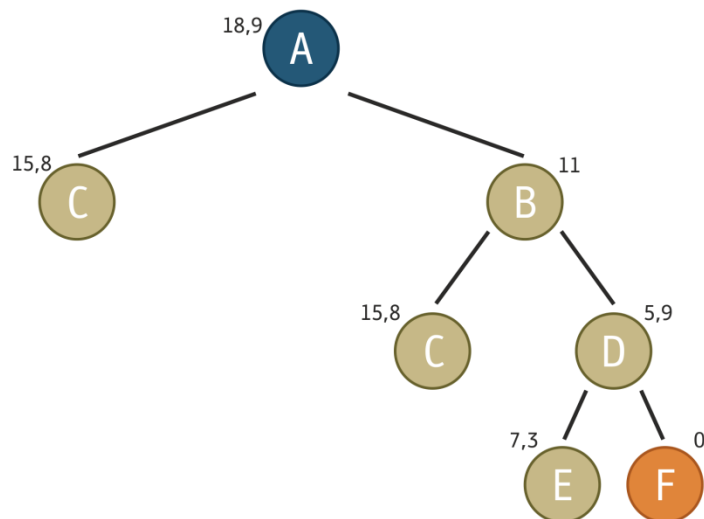
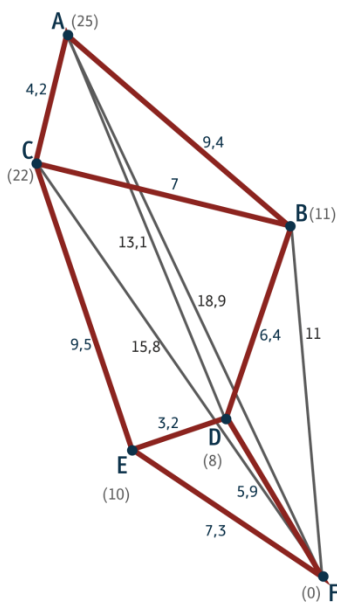
Εάν το μέτωπο δεν είναι άδειο,

Παράδωσε το πρώτο μονοπάτι της ουράς ως λύση.

ΤΕΛΟΣ

Στο σχήμα 1.24 δίνουμε μια μικρή περιγραφή της εφαρμογής της μεθόδου αναζήτησης BestFS για το πρόβλημα του πωλητή με τις 6 πόλεις (βλέπε [Σχήμα 1.1](#)), όταν αρχική κατάσταση είναι η πόλη A, στόχος η πόλη F και ευρετικό κριτήριο η Ευκλείδεια απόσταση. Δίπλα στους κόμβους του γραφήματος και του δένδρου αναζήτησης παρουσιάζονται οι Ευκλείδειες αποστάσεις των πόλεων του δικτύου από την πόλη-στόχο F.

Στην αρχή, ο αλγόριθμος δημιουργεί το αρχικό μέτωπο τοποθετώντας μέσα την κατάσταση A και την αντίστοιχη αρχική ουρά. Στη συνέχεια, υπολογίζεται η Ευκλείδεια απόσταση της πόλης A από την F και, δεδομένου ότι είναι η μοναδική κατάσταση στο μέτωπο, επιλέγεται ως η «καλύτερη», για να αποτελέσει γονικό κόμβο. Δημιουργούνται οι διάδοχοι κόμβοι, δηλαδή οι κόμβοι B και C, και τοποθετούνται μέσα στο μέτωπο στη θέση του γονικού κόμβου. Ακολουθεί ταξινόμηση των κόμβων του μετώπου και τα αντίστοιχα συμβαίνουν για την ουρά. Τέλος ενημερώνεται το κλειστό σύνολο. Στον επόμενο βήμα το μέτωπο περιέχει τις καταστάσεις B και C, οι οποίες ταξινομούνται βάσει των αποστάσεων αυτών από την F. Η πλησιέστερη είναι η B· άρα, επιλέγεται ως γονικός κόμβος δίνοντας τους διάδοχους κόμβους C και D. Τώρα το μέτωπο περιέχει τους κόμβους C, που ανήκει στη μερική λύση (A C), C, που ανήκει στη μερική λύση (A B C), και τον D. Ο D έχει τη μικρότερη απόσταση έναντι των δύο άλλων ισοτίμων C και, επομένως, επιλέγεται ως γονικός κόμβος. Η αναζήτηση συνεχίζεται με αυτήν τη λογική, μέχρι να βρει το στόχο και να επιστρέψει ως λύση το μονοπάτι (A B D F).



Σχήμα 1.24 Παράδειγμα εφαρμογής της BestFS για το πρόβλημα του περιοδεύοντος πωλητή, όπως παρουσιάζεται στο [σχήμα 1.1](#)

Δείτε [κινούμενη εικόνα 1.7 - Ανάπτυξη Δένδρου Αναζήτησης με την BestFS για το πρόβλημα του περιοδεύοντος πωλητή](#)

Δείτε [κινούμενη εικόνα 1.8 - Ανάπτυξη Δένδρου Αναζήτησης με την BestFS για το πρόβλημα των 8 πλακιδίων](#)

Προτερήματα της BestFS

- Συνήθως βρίσκει κάποια καλή λύση πιο γρήγορα από οποιαδήποτε άλλη μέθοδο. Η απόδοση εξαρτάται άμεσα από την ακρίβεια της λειτουργίας του ευρετικού μηχανισμού.
- Είναι πλήρης.

Μειονεκτήματα της BestFS

- Το μέτωπο αναζήτησης μεγαλώνει πολύ γρήγορα, γιατί ο αλγόριθμος κρατά όλες τις καταστάσεις, με αποτέλεσμα να απαιτείται μεγάλος χώρος για την αποθήκευσή του.
- Δεν εγγυάται ότι η λύση που θα βρεθεί είναι η καλύτερη όλων, γιατί σταματά στην πρώτη καλύτερη που θα συναντήσει, χωρίς να πραγματοποιεί εξαντλητική αναζήτηση για εύρεση πιθανής άλλης ακόμα καλύτερης.

1.5.3. Αναζήτηση με αναρρίχηση λόφων

Η μέθοδος της αναζήτησης με **Αναρρίχηση Λόφων** (Hill-Climbing - HC) αποτελεί μια εξειδίκευση της BestFS μεθόδου. Η πιο κλασική τεχνική που χρησιμοποιούν οι αλγόριθμοι αναρρίχησης λόφων είναι αυτή της πλέον απότομης ανάβασης (*steepest ascent*), βάσει της οποίας ο αλγόριθμος σε κάθε βήμα του αντικαθιστά τον τρέχοντα κόμβο με τον καλύτερο γειτονικό του. Η αναρρίχηση λόφων με απότομη ανάβαση έχει τα ακόλουθα χαρακτηριστικά:

- Η αναζήτηση γίνεται κατά μήκος ενός μόνο μονοπατιού στο χώρο αναζήτησης.
- Η αναζήτηση προχωρά μόνο από διαδοχικά καλύτερους κόμβους, κλαδεύοντας σε κάθε κύκλο αναζήτησης όλες τις καταστάσεις του μετώπου εκτός από την ευρετικά βέλτιστη (δηλαδή κάθε στιγμή το μέτωπο περιέχει μόνο μία κατάσταση) και μεταβαίνει στην τελευταία, μόνο αν έχει καλύτερη ευρετική τιμή από το γονέα της, διαφορετικά τερματίζει έχοντας βρει μία τοπικά βέλτιστη λύση.
- Αν υπάρχουν περισσότεροι του ενός καλύτεροι διάδοχοι κόμβοι, τότε η αναζήτηση επιλέγει τυχαία έναν από αυτούς.

Προφανώς ο αλγόριθμος αναρρίχησης λόφων με απότομη ανάβαση δεν είναι πλήρης, αλλά είναι πολύ γρήγορος και απαιτεί ελάχιστη μνήμη. Η διαφορά του με την αναζήτηση Καλύτερης-Πρώτης είναι ότι στην αναρρίχηση λόφων η ταξινόμηση και επιλογή γίνονται μεταξύ των διάδοχων κόμβων, ενώ στην Καλύτερη-Πρώτη η ταξινόμηση και επιλογή γίνονται σε όλους τους κόμβους του μετώπου.

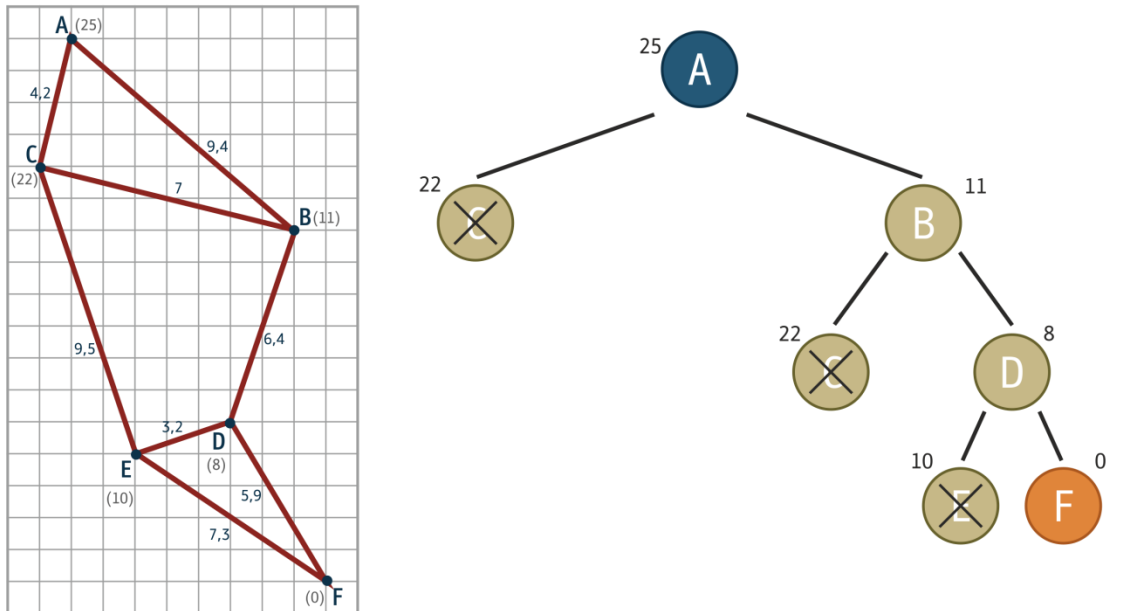
Αλγόριθμος μεθόδου αναζήτησης με Αναρρίχηση Λόφων

...

Εφάρμοσε τους τελεστές, για να παραγάγεις τις διάδοχες καταστάσεις που σχηματίζονται από την πρώτη κατάσταση του μετώπου.

Αφαίρεσε όλες τις καταστάσεις από το μέτωπο.
 Πρόσθεσε τις διάδοχες καταστάσεις στο άδειο μέτωπο και ταξινόμησέ τις βάσει ενός ευρετικού αλγορίθμου.
 ...

Στο σχήμα 1.25 φαίνεται το δένδρο αναζήτησης λύσης του προβλήματος του πωλητή με τις 6 πόλεις (βλέπε [Σχήμα 1.1](#)), όπως προκύπτει από την αναρρίχηση λόφων, όταν αρχική κατάσταση είναι η πόλη A, στόχος η πόλη B και ευρετικό κριτήριο είναι η απόσταση Manhattan κάθε κόμβου από τον κόμβο στόχο. Οι αποστάσεις Manhattan αποτυπώνονται δίπλα σε κάθε κόμβο.



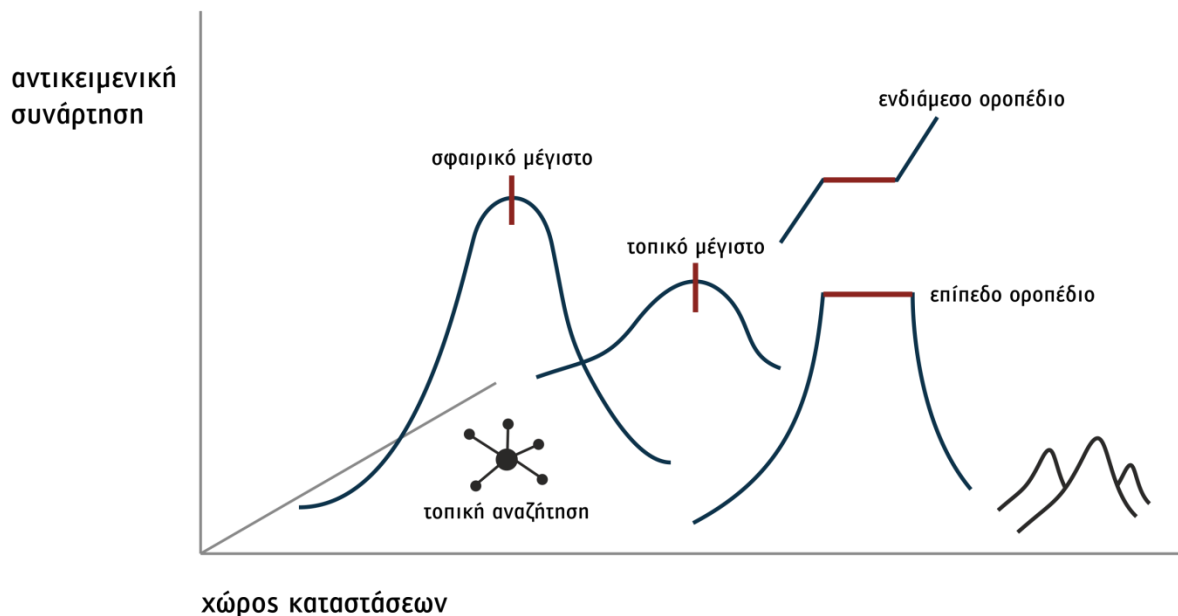
Σχήμα 1.25 Αναζήτηση λύσης με τη μέθοδο Αναρρίχησης Λόφων για το πρόβλημα του περιοδεύοντος πωλητή, όπως παρουσιάζεται στο [σχήμα 1.1](#)

Δείτε [κινούμενη εικόνα 1.9 - Ανάπτυξη Δένδρου Αναζήτησης με την HC για το πρόβλημα του περιοδεύοντος πωλητή](#)

Η αναρρίχηση λόφων λέγεται και **άπληστη τοπική αναζήτηση** (greedy local search), επειδή επιλέγει μια καλή γειτονική κατάσταση, χωρίς να σκέφτεται πού θα πάει στη συνέχεια. Όπως όλοι οι αλγόριθμοι άπληστης αναζήτησης, αποδίδει αρκετά καλά, κάνοντας συχνά πολύ γρήγορη πρόοδο προς μια λύση, επειδή συνήθως είναι πολύ εύκολο να βελτιώσει μια κακή κατάσταση. Δυστυχώς, συχνά η αναρρίχηση λόφων παγιδεύεται για τους εξής λόγους:

- τα τοπικά μέγιστα (local maxima),
- τα οροπέδια (plateau),
- τις κορυφογραμμές (ridges).

Η παγίδευση παρουσιάζεται, όταν υπάρχει μια θέση που δεν είναι στόχος, αλλά από εκεί καμία κίνηση δεν βελτιώνει την τρέχουσα κατάσταση ή η θέση οδηγεί σε πρώιμη σύγκλιση, δηλαδή εύρεση λύσης που δεν αποτελεί τη βέλτιστη λύση. Αυτό θα συμβεί, αν έχουμε φτάσει σε ένα τοπικό μέγιστο, ένα οροπέδιο ή μια κορυφογραμμή (βλέπε Σχήμα 1.26).



Σχήμα 1.26 Προβλήματα που μπορούν να δημιουργηθούν κατά την αναρρίχηση λόφων

Τοπικό μέγιστο: Είναι μια κατάσταση που είναι καλύτερη από όλες τις γείτονές της, αλλά δεν είναι καλύτερη από όσο κάποιες άλλες πιο μακριά που δεν μπορούν να προσεγγιστούν από την πορεία που έχει ακολουθηθεί. Όλες οι πιθανές επόμενες κινήσεις φαίνεται να οδηγούν σε χειρότερες καταστάσεις. Λύση σε αυτό είναι η οπισθοδρόμηση σε κάποια προηγούμενη κατάσταση και η συνέχιση της αναζήτησης σε διαφορετική κατεύθυνση· όμως, ο κλασικός αλγόριθμος αναρρίχησης λόφων δεν το επιτρέπει.

Οροπέδιο: Είναι μια επίπεδη έκταση του χώρου αναζήτησης στην οποία ένα σύνολο γειτονικών καταστάσεων έχουν την ίδια ευρετική τιμή. Ως εκ τούτου, δεν είναι δυνατόν να καθοριστεί η καλύτερη κατεύθυνση προς την οποία πρέπει να συνεχιστεί η αναζήτηση. Λύση είναι να διαμορφωθεί έτσι ο αλγόριθμος, ώστε να μπορεί να κάνει ένα μεγάλο άλμα προς κάποια κατεύθυνση και να προσπαθήσει να προσεγγίσει ένα νέο τμήμα του χώρου αναζήτησης.

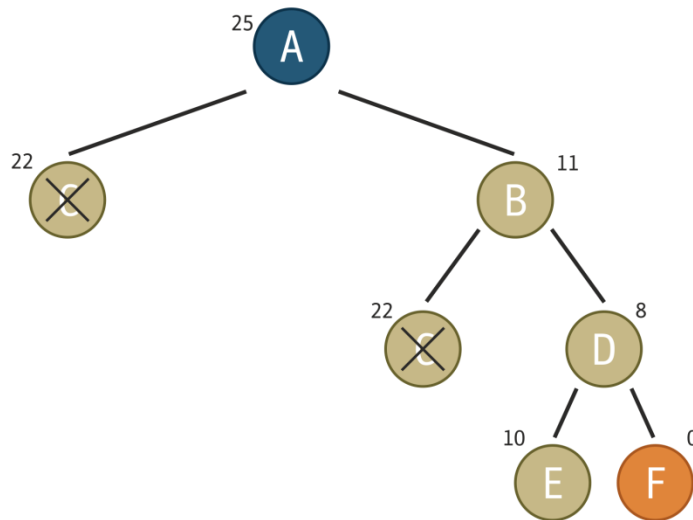
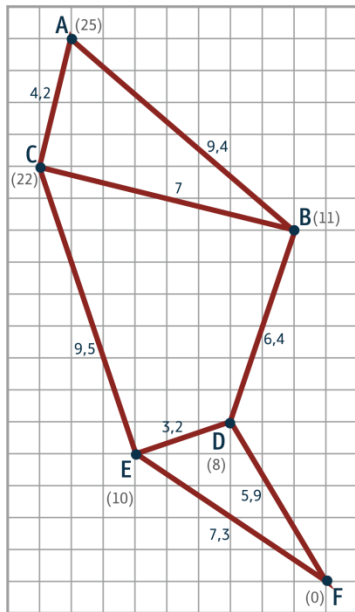
Κορυφογραμμές: Είναι μια περιοχή του χώρου αναζήτησης υψηλότερη από τις γύρω περιοχές (τοπική κορυφή), αλλά, αν η αναζήτηση βρίσκεται σε μια γειτονική κορυφογραμμή, δεν μπορεί να την προσεγγίσει με μια μοναδική κίνηση (ειδική περίπτωση τοπικών μεγίστων). Εδώ η εφαρμογή δύο ή περισσότερων βημάτων μετάβασης προς πολλές κατευθύνσεις ταυτόχρονα πρέπει να προηγηθεί του ευρετικού μηχανισμού για τον έλεγχο της καλύτερης πορείας.

Γενικές ενέργειες για την αντιμετώπιση παγιδεύσεων, όπως αυτές που αναφέραμε παραπάνω, είναι οι ακόλουθες:

- πλάγιες κινήσεις,
- στοχαστική αναζήτηση,
- τυχαία επιλογή καλών κινήσεων,
- τυχαίες επανεκκινήσεις,
- πλήρης αλγόριθμος.

Διάφορες παραλλαγές του αλγόριθμου αναρρίχησης λόφων χρησιμοποιούν τους παραπάνω τρόπους αποφυγής παγιδεύσεων, θυσιάζοντας λίγο από την ταχύτητα της αναζήτησης, για να αυξήσουν την πιθανότητα να βρεθεί λύση. Μία παραλλαγή είναι η **Εξαναγκασμένη Αναρρίχηση Λόφων** (Enforced Hill Climbing – EHC). Μια από τις πιο απλές είναι αυτή της EHC που καθιστά τον αλγόριθμο πλήρη. Στην EHC επιτρέπεται η οπισθοδρόμηση, με διατήρηση στο μέτωπο και των καταστάσεων του ακριβώς προηγούμενου

βήματος. Στο σχήμα 1.27 το παράδειγμα εφαρμογής είναι παραλλαγή του παραδείγματος του σχήματος 1.25, όπου σε κάθε βήμα κρατούνται όλα τα παιδιά του συγκεκριμένου βήματος για τις ανάγκες της οπισθοδρόμησης, που είναι επιτρεπτή μόνο κατά ένα βήμα, και στο επόμενο βήμα διαγράφονται για να αντικατασταθούν με τα παιδιά του επόμενου βήματος.



Σχήμα 1.27 Αναζήτηση λύσης με EHC για το πρόβλημα του περιοδεύοντος πωλητή

Επίσης δημοφιλής είναι και ο **αλγόριθμος προσομοιωμένης ανόπτωσης** (Simulated Annealing Algorithm) ο οποίος δίνει μία πιθανότητα μετάβασης σε χειρότερες καταστάσεις, αφήνοντας έτσι περιθώριο στην αναζήτηση να ξεφύγει από τοπικά βέλτιστα. Άλλος αλγόριθμος αναρρίχησης λόφων είναι ο **αλγόριθμος αναζήτησης με απαγορεύσεις** (Taboo Search Algorithm), όπου σε κάθε βήμα επέκτασης γίνεται πάντα μετάβαση στη καλύτερη διάδοχη κατάσταση, ακόμα και αν είναι χειρότερη από τη γονική, ενώ ταυτόχρονα η αναζήτηση αποφεύγει απαγορευμένες καταστάσεις αποθηκευμένες σε λίστα από την αρχή της αναζήτησης (παρόμοιας λειτουργικότητας με το κλειστό σύνολο, αλλά σταθερού μεγέθους).

1.5.4 Αναζήτηση Βασικής Επέκτασης και Οριοθέτησης

Η μέθοδος **αναζήτησης Βασικής Επέκτασης και Οριοθέτησης** (Basic Branch & Bound search - B&B) εφαρμόζεται σε προβλήματα όπου αναζητείται η λύση με το ελάχιστο κόστος. Η αναζήτηση δεν επισκέπτεται κόμβους του δένδρου αναζήτησης που οδηγούν σε λύσεις που υπερβαίνουν το μέγιστο επιτρεπόμενο κόστος.

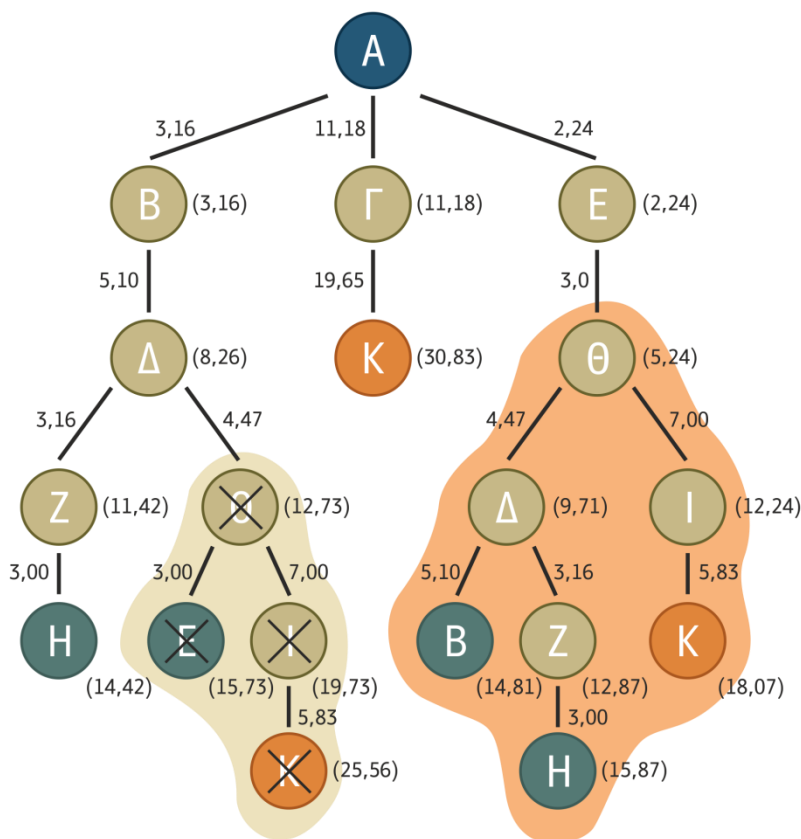
Αλγόριθμος αναζήτησης B&B

ΑΡΧΗ

-
- Εφάρμοσε τους τελεστές, για να παραγάγεις τις διάδοχες καταστάσεις που σχηματίζονται **από κάθε κατάσταση** του μετώπου καθιστώντας τις καταστάσεις-γονείς, και τοποθέτησέ τις στο μέτωπο, αφού πρώτα αφαιρέσεις όλες τις καταστάσεις-γονείς.
-

- **Ταξινομήσε το μέτωπο** βάσει ενός μέχρι τώρα κόστους αναζήτησης και **άφησε στο μέτωπο** μόνο αυτές τις καταστάσεις-παιδιά προς τις οποίες η διαδρομή δεν υπερβαίνει ένα μέγιστο επιτρεπόμενο κόστος.
 -
- ΤΕΛΟΣ

Στο σχήμα 1.28 δίνεται παράδειγμα εφαρμογής της μεθόδου αναζήτησης B&B κατά την εξαντλητική αναζήτηση λύσεων στο πρόβλημα του περιοδεύοντος πωλητή σε 10 πόλεις. Ως κριτήριο είναι το κόστος του μονοπατιού μέχρι τον τρέχοντα κόμβο. Από τα δύο υπο-δένδρα που σχηματίζονται κάτω από τους κόμβους Θ, η μέθοδος θα αποκόψει το αριστερό, γιατί το κόστος του μονοπατιού Α-Β-Δ-Θ είναι 12,73, ενώ του Α-Ε-Θ είναι μόνο 5,24.



Σχήμα 1.28 Εξαντλητική αναζήτηση B&B για το πρόβλημα του περιοδεύοντος πωλητή

1.5.5 Ακτινωτή μέθοδος αναζήτησης

Η **Ακτινωτή μέθοδος αναζήτησης** (Beam search) μπορεί να θεωρηθεί βελτίωση των μεθόδων αναζήτησης Αναρρίχησης Λόφου και Καλύτερης-Πρώτης.

Διαφέρει αφενός από την Αναρρίχηση Λόφων στο ότι προχωρά ανά επίπεδο και όχι σε ένα μονοπάτι με ολοένα καλύτερες καταστάσεις και αφετέρου από την αναζήτηση Καλύτερης-Πρώτης στο ότι, καθώς προχωρά σε βάθος, βασίζεται στην ποιότητα των κόμβων κάθε επιπέδου επισκεπτόμενη μόνο τον καλύτερο κόμβο κάθε επιπέδου που εντοπίζει η ευρετική συνάρτηση, αγνοώντας τους υπόλοιπους κόμβους. Με αυτόν τον τρόπο μειώνεται σημαντικά ο αριθμός των κόμβων που επισκέπτεται.

Τα χαρακτηριστικά της μεθόδου είναι τα ακόλουθα:

- Ο κύκλος αναζήτησης εξελίσσεται από επίπεδο σε επίπεδο.
- Κινείται προς τα κάτω διατηρώντας τους καλύτερους N κόμβους μόνο σε κάθε επίπεδο. Το N ονομάζεται *πλάτος της ακτινωτής αναζήτησης*.
- Άλλοι κόμβοι παραβλέπονται.

Αλγόριθμος ακτινωτής μεθόδου αναζήτησης

ΑΡΧΗ

...

Έως ότου η αναζήτηση καταλήξει στον κόμβο-σκοπό ή το μέτωπο αδειάσει,

Εφάρμοσε τους τελεστές μετάβασης, για να σχηματίσεις τους διάδοχους κόμβους κάθε κόμβου του μετώπου.

Αδειασε το μέτωπο από τους γονικούς κόμβους.

Ταξινόμησε τους διάδοχους κόμβους βάσει ενός ευρετικού κριτηρίου και **τοποθέτησε N το πλήθος από αυτούς στο άδειο μέτωπο.**

Τέλος έως ότου

...

ΤΕΛΟΣ

Στο σχήμα 1.29, έχει αποτυπωθεί η ανά επίπεδο πορεία της ακτινωτής αναζήτησης για το πρόβλημα του περιοδεύοντος σε 10 πόλεις πωλητή, όταν το κριτήριο είναι το μήκος του μονοπατιού μέχρι τον τρέχοντα κόμβο. Το πλάτος της αναζήτησης είναι 2 και ξεκινά από το επίπεδο 0 όπου το μέτωπο περιέχει μόνο τον κόμβο της αρχικής κατάστασης.

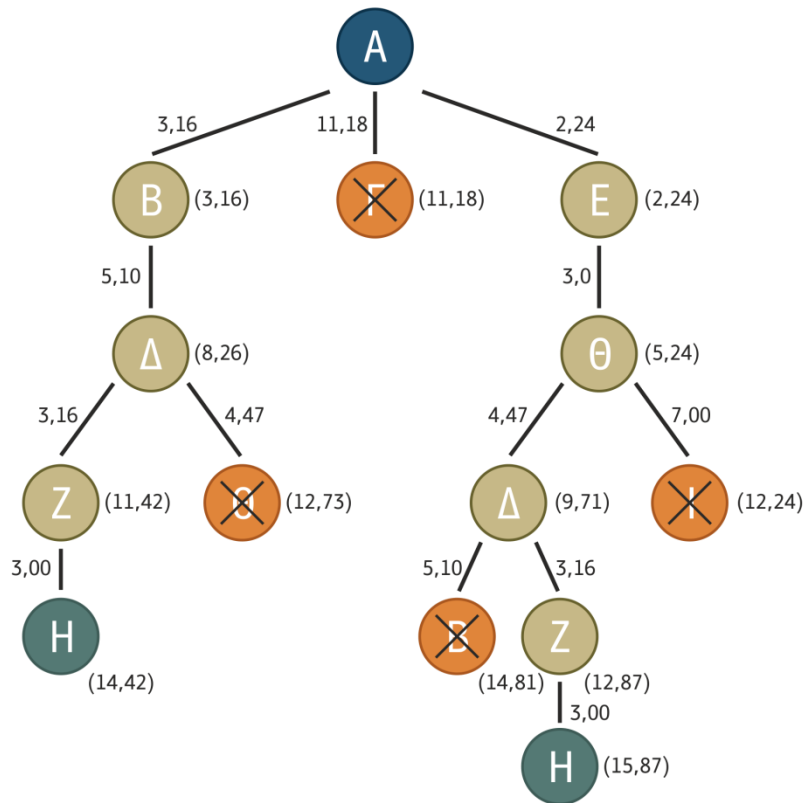
Στο επίπεδο 1 που διαμορφώνεται από τον επόμενο κύκλο λειτουργίας δημιουργούνται όλοι οι διάδοχοι κόμβοι του προηγούμενου επιπέδου και παραμένουν στο μέτωπο οι E και B ως οι δύο καλύτεροι.

Στο επίπεδο 2 δημιουργούνται οι διάδοχοι κόμβοι των E και B και παραμένουν στο μέτωπο όλοι, γιατί είναι μόνο δύο, οι Δ και Θ .

Στο επίπεδο 3 δημιουργούνται οι διάδοχοι κόμβοι Z , Θ , I , Δ και παραμένουν οι Z και Δ ως οι δύο καλύτεροι.

Στο επίπεδο 4 δημιουργούνται οι διάδοχοι κόμβοι H , B , Z και παραμένουν οι H , Z ως οι δύο καλύτεροι.

Στο επίπεδο 5 ο κόμβος H δεν έχει διάδοχο κόμβο, γιατί θα δημιουργείτο κύκλος, ενώ ο Z έχει διάδοχο κόμβο τον H που μπαίνει στο μέτωπο. Ο H αυτός δεν έχει διάδοχο κόμβο, γιατί θα οδηγείτο σε κύκλο και έτσι η αναζήτηση ολοκληρώνεται χωρίς λύση, παρότι το πρόβλημα έχει τρεις λύσεις.



Σχήμα 1.29 Αποτυχημένη αναζήτηση με ακτινωτή μέθοδο και κριτήριο το μήκος του μονοπατιού μέχρι τον τρέχοντα κόμβο

1.5.6 Μέθοδος αναζήτησης Άλφα-Άστρο

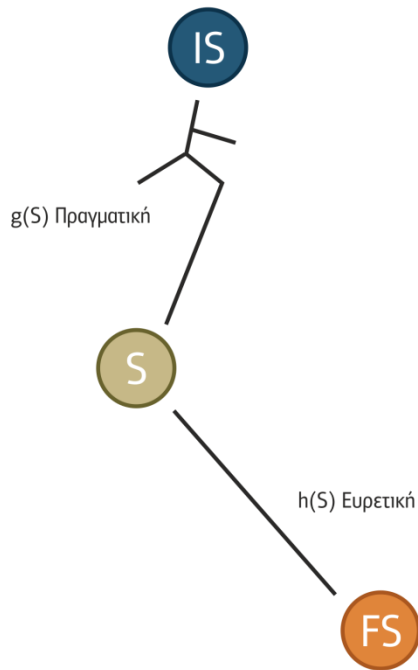
Η μέθοδος αναζήτησης Άλφα-Άστρο (A*) είναι κατά βάσει BestFS, αλλά με ευρετική συνάρτηση:

$$F(S) = g(S) + h(S)$$

Όπου:

- η $g(S)$ δίνει την απόσταση της S από την αρχική κατάσταση, η οποία είναι πραγματική και γνωστή, και
- η $h(S)$ δίνει την εκτίμηση της απόστασης της S από την τελική κατάσταση μέσω μιας ευρετικής συνάρτησης, όπως ακριβώς στον BestFS.

Για την A* ισχύει ότι, αν για κάθε κατάσταση η τιμή $h(S)$ είναι μικρότερη ή το πολύ ίση με την πραγματική απόσταση της S από την τελική κατάσταση, τότε η A* βρίσκει πάντα τη βέλτιστη λύση.



Σχήμα 1.30 Η πορεία της A* από μια αρχική κατάσταση IS μέχρι μια τελική κατάσταση FS, όταν η αναζήτηση βρίσκεται στην κατάσταση S

Ακολουθεί ο αλγόριθμος της A*

ΑΡΧΗ

...

- Εφαρμοσε τους τελεστές μετάβασης, για να υπολογίσεις τις διάδοχες καταστάσεις της καλύτερης κατάστασης του μετώπου καθιστώντας την γονικό κόμβο.
- AN υπάρχουν διάδοχες καταστάσεις,
 - Αφαίρεσε το γονικό κόμβο από το μέτωπο.
 - Πρόσθεσε στο μέτωπο **μόνο** εκείνες από τις διάδοχες καταστάσεις προς τις οποίες το κόστος της διαδρομής είναι μικρότερο από το κόστος ίδιων καταστάσεων που βρίσκονται ήδη στο μέτωπο από προηγούμενα βήματα αναζήτησης.
 - **Αφαίρεσε** τις καταστάσεις του μετώπου από παλαιότερα βήματα σε περίπτωση κατά την οποία προστέθηκαν στο μέτωπο νέες ίδιες με αυτές διάδοχες καταστάσεις.
 - **Ταξινόμησε όλες** τις καταστάσεις του μετώπου **συνυπολογίζοντας** (με υπερτίμηση) το μέχρι τώρα κόστος αναζήτησης και (με υποτίμηση) το κόστος που υπολείπεται μέχρι την κατάσταση-στόχο.
- Τέλος AN

...

ΤΕΛΟΣ

Για παράδειγμα, αν στο πρόβλημα των 8 πλακιδίων η A* εφαρμόσει την ευρετική συνάρτηση:

$$f(S) = g(S) + h(S)$$

όπου:

$g(S)$ = το βάθος του κόμβου της κατάστασης S στο δένδρο αναζήτησης
 $h(S)$ = ο αριθμός των κόμβων εκτός θέσης στη δεδομένη κατάσταση S ,

και μια τρέχουσα κατάσταση S με αριθμό κόμβων εκτός θέσης 4 όπως παρακάτω:

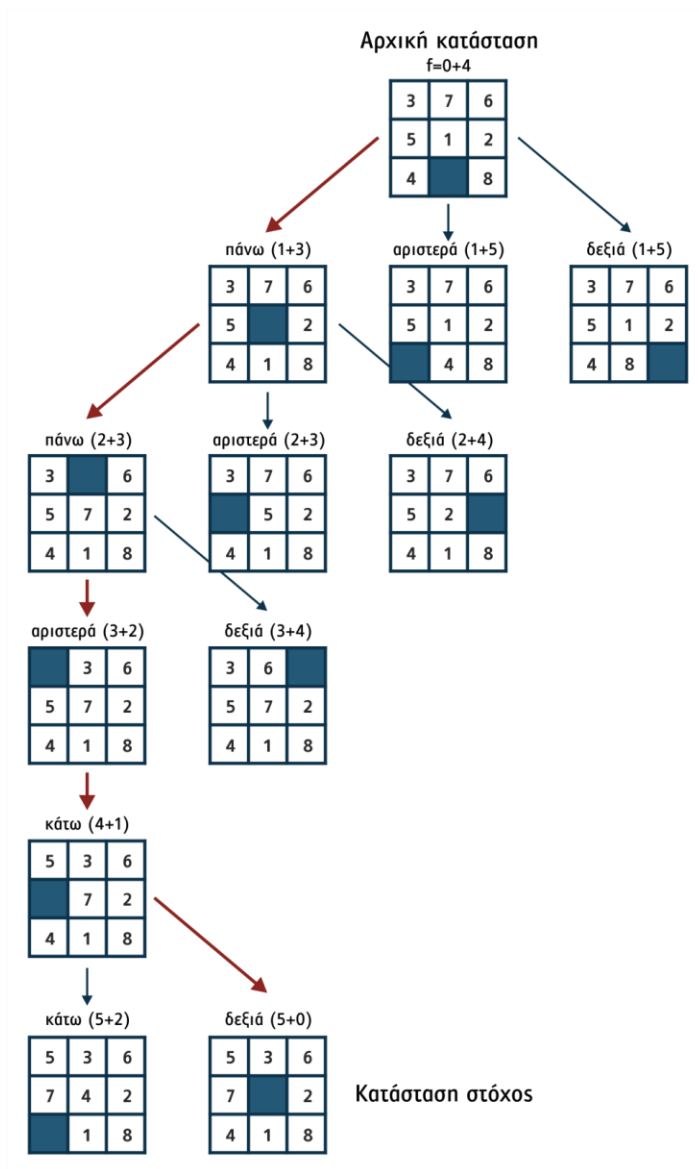
Τρέχουσα κατάσταση	3	7	6
	5	1	2
	4		8

Τελική κατάσταση	5	3	6
	7		2
	4	1	8

τότε, αν η τρέχουσα κατάσταση S είναι η αρχική (με βάθος 0), ο κόμβος που αντιστοιχεί θα έχει ευρετική τιμή:

$$f(\text{initial_node}) = 0 + 4 = 4$$

και το δένδρο που θα σχηματίσει η A^* δίνεται στο επόμενο σχήμα 1.31 όπου με κόκκινα βέλη αποτυπώνεται το μονοπάτι-λύση.



Σχήμα 1.31 Δένδρο αναζήτησης της A* για τα το πρόβλημα των 8 πλακιδίων

Δείτε [κινούμενη εικόνα 1.10 - Ανάπτυξη Δένδρου Αναζήτησης με την A* για το πρόβλημα των 8-puzzles](#)

Βιβλιογραφία/Αναφορές

- Russell, S. J. & Norvig, P. (2007). *Τεχνητή Νοημοσύνη: μια σύγχρονη προσέγγιση*. Αθήνα:Κλειδάριθμος.
- Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κόκκορας, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστημίου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Αντιστοιχίστε τους όρους:

1. Δένδρο Αναζήτησης	A. Είναι το σύνολο όλων των καταστάσεων ενός προβλήματος που είναι προσβάσιμες από την αρχική κατάσταση.
2. Χώρος Αναζήτησης	B. Είναι το σύνολο των έγκυρων καταστάσεων στις οποίες μπορεί να βρεθεί ένα πρόβλημα κατά την εξέλιξη του κόσμου του.
3. Χώρος Καταστάσεων	C. Είναι η δομή που απεικονίζει την πορεία που ακολουθεί ένας αλγόριθμος αναζήτησης κατά την επίλυση ενός προβλήματος.

Απάντηση/Λύση

Η σωστή αντιστοίχιση είναι: 1→C, 2→B, 3→A

Κριτήριο αξιολόγησης 2

Η λύση ενός προβλήματος ορίζεται από τα παρακάτω (επιλέξτε αυτά που θεωρείτε ορθότερα):

- A. την τελική κατάσταση-στόχο και τη μέθοδο αναζήτησης που εφαρμόστηκε, για να βρεθεί αυτή
- B. την ακολουθία των καταστάσεων από τις οποίες διέρχεται το πρόβλημα στην πορεία του από μια κατάσταση αρχική προς μια τελική κατάσταση-στόχο
- Γ. την ακολουθία των τελεστών που εφαρμόζονται, για να βρεθεί το πρόβλημα από την αρχική κατάσταση σε μια τελική κατάσταση-στόχο, χρησιμοποιώντας μια συγκεκριμένη μέθοδο αναζήτησης
- Δ. τη μέθοδο αναζήτησης που εφαρμόζεται, για να βρεθεί το πρόβλημα από την αρχική κατάσταση σε μια τελική κατάσταση-στόχο διασχίζοντας το χώρο των επιτρεπτών καταστάσεων

Απάντηση/Λύση

Σωστά είναι τα B και C.

Κριτήριο αξιολόγησης 3

Ένας εξαντλητικός ευρετικός αλγόριθμος αναζήτησης, για ένα πρόβλημα που έχει περισσότερες της μιας λύσεις, τι από τα παρακάτω βρίσκει;

- A. όλες ανεξαιρέτως τις υπάρχουσες λύσεις
- B. όλες τις βέλτιστες λύσεις
- C. μόνο την καλύτερη από τις υπάρχουσες λύσεις
- D. όλες τις υπάρχουσες λύσεις που δεν έχει αποκόψει η ευρετική μέθοδος αναζήτησης

Απάντηση/Λύση

Η σωστή απάντηση είναι το D.

Κριτήριο αξιολόγησης 4

Ένα πρόβλημα ορίζεται πλήρως από τα παρακάτω (επιλέξτε):

- A. την είσοδο (I) του προβλήματος
- B. όλες τις πιθανές εισόδους (I) του προβλήματος
- C. όλους τους πιθανούς στόχους (G) του προβλήματος
- D. τους τελεστές μετάβασης (T) του προβλήματος
- E. την ακολουθία των καταστάσεων (S) που οδηγεί στη λύση του προβλήματος
- F. το χώρο επιτρεπτών καταστάσεων (S) του προβλήματος

Απάντηση/Λύση

Σωστή απάντηση είναι τα A, C, D, F.

ΚΕΦΑΛΑΙΟ 2 - Αναπαράσταση Γνώσης

Σύνοψη

Στο χώρο της επίλυσης προβλημάτων, κυρίως του πραγματικού κόσμου, η καταλληλότητα της επιλεγμένης μεθόδου για την αναπαράσταση της γνώσης που υπάρχει σχετικά με ένα πρόβλημα είναι θεμελιώδους σημασίας, όταν αυτό πρόκειται να επιλυθεί μέσω συστημάτων διαχείρισης της γνώσης, όπως είναι, για παράδειγμα, τα έμπειρα συστήματα. Στο συγκεκριμένο κεφάλαιο αναλύεται η έννοια της γνώσης και παρουσιάζονται οι βασικές λογικές και δομημένες μέθοδοι αναπαράστασης γνώσης, μεταξύ των οποίων συγκαταλέγονται ο κατηγορηματικός λογισμός πρώτης τάξης, τα σημασιολογικά δίκτυα, τα πλαίσια, τα σενάρια και οι κανόνες. Επίσης, ιδιαίτερη αναφορά γίνεται στους εννοιολογικούς χάρτες και στις οντολογίες.

Προαπαιτούμενη γνώση

Δεν υπάρχει προαπαιτούμενη γνώση.

2.1 Γνώση

Η γνώση αποτελεί ακρογωνιαίο λίθο της ανθρώπινης συμπεριφοράς, της οποίας τα ευφυή στοιχεία προσπαθεί ο άνθρωπος να μεταδώσει σε μια υπολογιστική μηχανή. Στο προηγούμενο [κεφάλαιο 1](#), αναφερθήκαμε πολλές φορές στον όρο ευφυΐα. 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.

Έχουν δοθεί πολλοί ορισμοί σχετικά με την ευφυΐα, αλλά θεωρείται σκοπιμότερο να περιγραφεί σύμφωνα με τα χαρακτηριστικά της:

Ευφυΐα (intelligence) είναι η συνεχής διαδικασία της συγκέντρωσης, εξέτασης, ταξινόμησης και διερμηνείας πληροφοριών.

Με τον όρο *πληροφορία* εννοούμε το σύνολο όλων των γεγονότων, των σχημάτων, των δεδομένων και των όρων που έχει κάποιος επινοήσει ή συγκεντρώσει, για να δημιουργήσει μία εσωτερική αναπαράσταση ή μοντέλο για τον κόσμο. Με τον όρο *διερμηνεία*, εννοούμε την επεξεργασία των πληροφοριών αυτών για επίλυση προβλημάτων, λήψη αποφάσεων και άντληση νέων πληροφοριών.

Για να οδηγήσει η πληροφορία σε μια ορθή διερμηνεία της, ο τρόπος με τον οποίο παρουσιάζεται μια πληροφορία πρέπει να καθιστά σαφές το νόημά της· για να το επιτύχει αυτό, συνήθως επιβάλλεται να παρουσιάζεται η πληροφορία τοποθετημένη μέσα στο εννοιολογικό της πλαίσιο. Για παράδειγμα, η παρακάτω αιγυπτιακή ιερογλυφική απεικόνιση έχει διερμηνευτεί από τους αρχαιολόγους, διότι τη συνέδεσαν εννοιολογικά με άλλα σύγχρονα ευρήματά της:



["Egyptian - Heart Scarab with Deceased Adoring Osiris - Walters 4283 - Impression Detail"](#) by [Walters Art Museum](#) in the Public Domain

Εικόνα 2.1 [Παράδειγμα πληροφορίας τοποθετημένης σε εννοιολογικό πλαίσιο](#)

Αντίθετα, τα ευρήματα των μονολιθικών ανθρωπόμορφων στηλών *moai* στη Χιλιανή Πολυνησία, που απεικονίζονται στην παρακάτω φωτογραφία, δεν έχουν ακόμα διερμηνευτεί πλήρως, διότι δεν έχει καταστεί δυνατό να τοποθετηθούν σε κάποιο γνωστό εννοιολογικό πλαίσιο.



"Ahu-Akivi-1" by Ian Sewell available under CC BY-SA 3.0

Εικόνα 2.2 [Παράδειγμα πληροφορίας μη τοποθετημένης σε εννοιολογικό πλαίσιο](#)

Πρόβλημα στη διερμηνεία της πληροφορίας μπορεί να δημιουργήσουν τα συμφραζόμενα, δηλαδή πόσες διαφορετικές ερμηνείες μπορεί να έχει αυτή· για παράδειγμα, η αγγλική λέξη 'glass' μπορεί να χρησιμοποιηθεί, για να εκφράσει το ποτήρι, το τζάμι, τα βιτρό ή τα γυαλιά όρασης ή ηλίου.

Σε μια τέτοια περίπτωση, το εννοιολογικό πλαίσιο θα μπορούσε να χρησιμοποιηθεί, για να οδηγήσει στη σωστή διερμηνεία, αλλά αυτό δεν είναι σίγουρο ότι θα είναι πάντα διαθέσιμο. Η ευκολότερη αντιμετώπιση της ύπαρξης συμφραζομένων είναι η πληροφορία να συνοδεύεται με περισσότερα στοιχεία που θα καθιστούν σαφές το νόημά της, π.χ. sun glasses.



Εικόνα 2.3 Συμφραζόμενα της έννοιας ‘glass’

Επειδή πληροφορίες που εμπεριέχουν ασάφειες, δηλαδή το νόημά τους είναι διαφορούμενο, δεν μπορούν να οδηγήσουν σε μια συγκεκριμένη διερμηνεία, δεν μπορεί να ελεγχθεί η εγκυρότητά τους, ώστε να αποτελέσουν γνώση. Κλασικό παράδειγμα είναι ο διαφορούμενος χρησμός της Πυθίας, όπου ανάλογα με το πού θα μπει το κόμμα κατά την εκφορά του προφορικού λόγου αλλάζει και το νόημα του χρησμού:

“*Ηξεις αφήξεις,
ούκ εν πολέμω θνήξεις*”

“*Ηξεις αφήξεις ούκ,
εν πολέμω θνήξεις*”

Τέλος, σημαντική είναι η αναπαράσταση μιας πληροφορίας για τη διερμηνεία της. Πρωταρχικός στόχος της έρευνας σχετικά με το τι είναι ευφυΐα και του πώς μπορεί να μεταδοθεί σε μια μηχανή είναι η κατανόηση του τρόπου αναπαράστασης των πιστοποιημένων πληροφοριών που κατέχουμε εσωτερικά στη μνήμη μας ως γνώσεις.

Γενικά, **αναπαράσταση** (representation) είναι το σύνολο των συντακτικών και σημασιολογικών παραδοχών που συγκροτούν μια οντολογία η οποία καθιστά δυνατή την περιγραφή ενός κόσμου ή μιας συγκεκριμένης κατηγορίας πραγμάτων με χρήση μιας συγκεκριμένης γλώσσας, η οποία διαθέτει το δικό της συντακτικό και τη δική της σημασιολογία.

Πρέπει, επίσης, να διευκρινιστεί ότι ο όρος *αναπαράσταση* διαφέρει από τον όρο *περιγραφή* με τον οποίο εννοούμε τη χρήση των παραδοχών για την αναπαράσταση ενός συγκεκριμένου στοιχείου μιας κατηγορίας. Η αναπαράσταση αντιστοιχεί στον προσδιορισμό μιας σύνταξης μέσω της οποίας περιγράφεται μια κατηγορία στοιχείων, ενώ η περιγραφή αφορά ένα αποτυπωμένο, μέσω αυτής της σύνταξης, στοιχείο της κατηγορίας σε κάποιο συγκεκριμένο στιγμιότυπο.

Η γνώση διαφέρει ως έννοια και από τα δεδομένα και από την πληροφορία (βλέπε Σχήμα 2.1).

Δεδομένο (data) είναι μία μετρήσιμη ή υπολογίσιμη τιμή μιας ιδιότητας.

Πληροφορία (information) είναι το δεδομένο στο οποίο δίνεται ένα πλαίσιο αναφοράς και μία μορφοποίηση.

Γνώση (knowledge) είναι οι πληροφορίες που έχουν πιστοποιηθεί ως προς την ποιότητά τους.

Σοφία (wisdom) καλείται η γνώση επί της γνώσης, δηλαδή η *μετα-γνώση*, που είναι απαραίτητη για τη σωστή χρήση της γνώσης.



Σχήμα 2.1 Η πυραμίδα της γνώσης

2.1.1 Είδη Γνώσης

Η γνώση που κατέχουμε στη μνήμη μακράς διάρκειας είναι δύο ειδών:

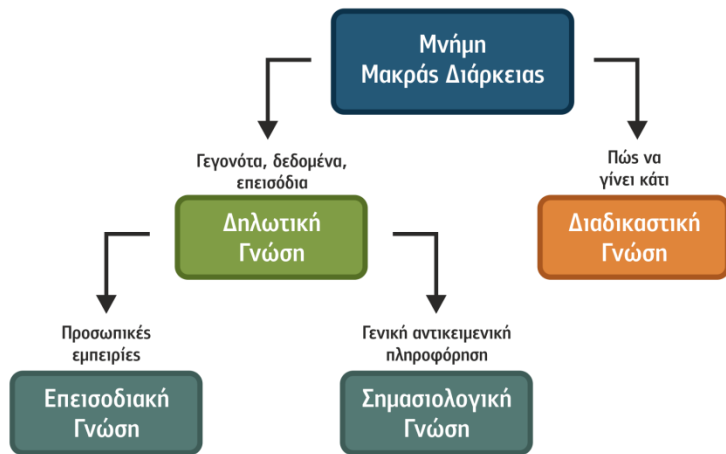
- **δηλωτική γνώση** (declarative knowledge) δηλαδή γνώση που έχουμε για τα αντικείμενα (*objects-data*), τα γεγονότα (*facts*) και τα συμβάντα (*events*),
- **διαδικαστική γνώση** (procedural knowledge) δηλαδή γνώση την οποία έχουμε για τις δεξιότητες που απαιτούνται, για να εκτελέσουμε μια εργασία ή να διεκπεραιώσουμε μια διαδικασία, και η οποία είναι αποτέλεσμα άμεσων αλλά και έμμεσων διαδικασιών μάθησης.

Παραδείγματα διαδικαστικής γνώσης:

- Για να φας σε ένα εστιατόριο, πρέπει να έχεις χρήματα να πληρώσεις.
- Όταν οδηγείς ποδήλατο, πρέπει να κρατάς την ισορροπία σου.
- Ενέργειες που πρέπει να κάνεις, για να συναρμολογήσεις έναν Η/Υ.

Η δηλωτική γνώση μπορεί περαιτέρω να διαχωριστεί σε

- **σημασιολογική γνώση** (semantic knowledge), που αφορά γενική γνώση για τις έννοιες που αντιπροσωπεύουν τα αντικείμενα (οντότητες) ενός κόσμου και τις μεταξύ τους σχέσεις, και
- **επεισοδιακή γνώση** (episodical knowledge), που αφορά προσωπικές εμπειρίες και βιωμένα γεγονότα στο παρελθόν και είναι οργανωμένη χρονικά και χωρικά σε επεισόδια σε σειριακή μορφή και όχι σε έννοιες ή σχέσεις.



Σχήμα 2.2 Ανάλυση των ειδών γνώσης

Παραδείγματα επεισοδιακής γνώσης:

- το όνομα του κατοικίδιου ζώου σας,
- ο γάμος της καλύτερης φίλης σας,
- η τιμωρία που σας επέβαλε ο δάσκαλός σας στην πέμπτη δημοτικού,
- το μέρος όπου ήσασταν όταν μάθατε για την καταστροφή Δίδυμων Πύργων.

Παραδείγματα δηλωτικής σημασιολογικής γνώσης:

- η κατανόηση της διαφοράς μεταξύ ενός σκύλου και μιας γάτας,
- η γνώση ότι η καταστροφή των Δίδυμων Πύργων έγινε στις 11 Σεπτεμβρίου 2001,
- η ικανότητα κάποιου να συνδέσει τα γράμματα με τους ήχους τους.

Η γνώση, δηλωτική ή διαδικαστική, διακρίνεται σε **γενική γνώση** (general knowledge, common sense) και **ειδική γνώση** (domain-specific knowledge) για ένα συγκεκριμένο θέμα που κατέχουμε. Π.χ. ξέρουμε γενικά ότι τα λεωφορεία μεταφέρουν επιβάτες από μέρος σε μέρος ή ότι ο πατέρας είναι μεγαλύτερος σε ηλικία από τον γιο του όμως ότι «το λεωφορείο Α8 πηγαίνει στο Μαρούσι» αποτελεί ειδική γνώση την οποία διαθέτουν κάποιοι που μετακινούνται προς το Μαρούσι, αλλά όχι απαραίτητα κάποιοι άλλοι που δεν έχουν επισκεφτεί ποτέ το Μαρούσι και ούτε ενδιαφέρονται να το κάνουν.

Παραδείγματα δηλωτικής γνώσης:

Γενική:

Οι γονείς είναι μεγαλύτεροι από τα παιδιά τους.
 Η λέξη «παιδί» είναι ουσιαστικό, ενώ το «ωραίος» είναι επίθετο.

Ειδική:

Το ΤΕΙ-Α είναι στο Αιγάλεω.
 Η Μαρία είναι ψηλή.

Παραδείγματα διαδικαστικής γνώσης:

Γενική:

Για να πάρεις πτυχίο, πρέπει να διαβάσεις.
 Για τον προσδιορισμό ενός ουσιαστικού προσθέτουμε μπροστά ένα επίθετο.

Ειδική:

Για να έχεις πρόσβαση στο Γυμναστήριο του ΤΕΙ Αθήνας, πρέπει να έχεις κάρτα μέλους.
 Πάτησε escape, για να βγεις από το πρόγραμμα.

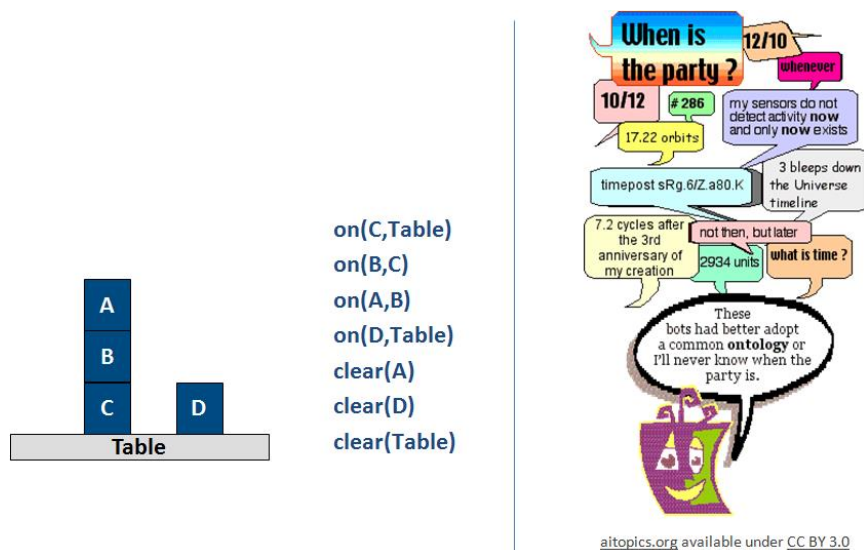
2.2 Αναπαράσταση Γνώσης

Ο καλύτερος τρόπος να εκφράσουμε τη γνώση είναι η φυσική γλώσσα. Για να γίνει, όμως, επεξεργάσιμη από έναν υπολογιστή, χρειάζεται να επιλεγεί πρώτα κατάλληλος τρόπος αποτύπωσής της μέσω κάποιου είδους αναπαράστασης και ακολούθως ένας φορμαλισμός, δηλαδή μιας τεχνητή γλώσσα με το δικό της συντακτικό και τη δική της σημασιολογία, προκειμένου η αναπαράσταση αυτή να καταστεί κατανοητή από τον υπολογιστή.

Δεδομένης της διαφοράς μεταξύ γνώσης, δεδομένων και πληροφοριών, είναι ευνόητο ότι οι μέθοδοι αναπαράστασης της γνώσης πρέπει να διαφέρουν από τις μεθόδους μορφοποίησης των δεδομένων και των πληροφοριών.

Η **αναπαράσταση της γνώσης** (knowledge representation) που χρησιμοποιείται για την επίλυση ενός προβλήματος μέσα στο χώρο της ΤΝ πρέπει να διαθέτει ένα μονοσήμαντο και τυποποιημένο συμβολισμό που:

- να δίνει μία ακριβή διερμηνεία χωρίς συμφραζόμενα.
- να μπορεί να συνδυαστεί κατάλληλα με μηχανισμούς εξαγωγής συμπερασμάτων (στρατηγική αναζήτησης - συλλογιστική).



Εικόνα 2.4 Δυο διαφορετικές προσεγγίσεις έκφρασης της γνώσης - <http://aitopics.org/>

Στην περίπτωση των παραδειγμάτων της εικόνας 2.4, στο δεξί παράδειγμα, που αφορά διαθέσιμη γνώση για την ώρα διεξαγωγής ενός πάρτι, υπάρχουν πολλοί διαφορετικοί τρόποι αναπαράστασης: ωστόσο, όλοι είναι ακατάλληλοι προς χρήση, καθώς άλλοι είναι ασαφείς (10/12, not the but later)) και άλλων η σύνταξη δε διευκολύνει τους μηχανισμούς εξαγωγής συμπερασμάτων. Στο αριστερό παράδειγμα της εικόνας, αυτό της γραφικής αναπαράστασης του κόσμου των κύβων, η αναπαράσταση δεν είναι κατάλληλη, δεδομένου ότι μια αναπαράσταση πρέπει να είναι συμβολική, για να μπορεί να συνδυαστεί κατάλληλα με μηχανισμούς εξαγωγής συμπερασμάτων. Η μεσαία αναπαράσταση είναι αντιθέτως κατάλληλη, διότι είναι συμβολική και παράλληλα, ορίζει σαφώς τις σχέσεις μεταξύ των κύβων, παρέχοντας τη δυνατότητα μιας μονοσήμαντης διερμηνείας της γνώσης που αναπαρίσταται.

2.2.1 Χαρακτηριστικά μιας «καλής» αναπαράστασης γνώσης

Για να γίνει σαφής μία μεταβολή του κόσμου ενός προβλήματος, πρέπει η επιλεγμένη αναπαράσταση της γνώσης να διαθέτει ένα σύνολο χαρακτηριστικών που θα την κάνουν αξιοποιήσιμη από έναν υπολογιστή. Θεωρητικά, από τη στιγμή κατά την οποία ένα πρόβλημα έχει αναπαρασταθεί κατάλληλα, τότε έχει σχεδόν λυθεί!

Αυτά τα χαρακτηριστικά αφορούν τη σύνταξή της, που πρέπει

- να είναι
 - κατανοητή, δηλαδή να δείχνει με μια ματιά το τι συμβαίνει,
 - πλήρης και σύντομη,
 - γρήγορη στην αναζήτηση,
 - εύκολα επεξεργάσιμη.
- να κάνει εμφανή την αλληλεπίδραση μεταξύ των οντοτήτων του προβλήματος.
- να έχει διαθέσιμες αλλά μη εμφανιζόμενες λεπτομέρειες.

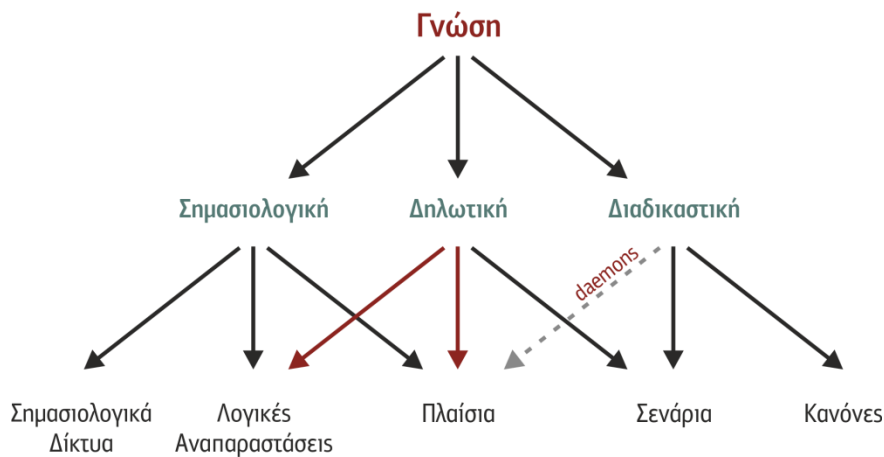
2.2.2 Μέθοδοι Αναπαράστασης Γνώσης

Για την αναπαράσταση της γνώσης, είναι απαραίτητο να χρησιμοποιείται μία μονοσήμαντη και τυποποιημένη σύνταξη, ένας φορμαλισμός που, όπως προείπαμε, πρέπει να δίνει μια ακριβή αναπαράσταση με νόημα (κατανόηση εννοιολογικού πλαισίου), χωρίς συμφραζόμενα, με κατανοητή πρόθεση, χωρίς ασάφειες, με δυνατότητα να συνδυαστεί κατάλληλα με μηχανισμούς εξαγωγής συμπερασμάτων (στρατηγική αναζήτησης - συλλογιστική).

Σημαντικότερες μέθοδοι αναπαράστασης που έχουν αναπτυχθεί στο πλαίσιο της ΤΝ είναι:

- *Σχήματα Λογικής Αναπαράστασης*
 - προτασιακός λογισμός,
 - κατηγορηματικός λογισμός,
 - κατηγορηματικός λογισμός πρώτης τάξης.
- *Δομημένες Μορφές Αναπαράστασης Γνώσης*
 - σημασιολογικά δίκτυα,
 - εννοιολογικοί χάρτες,
 - πλαίσια,
 - σενάρια.
- *Κανόνες*

Ο διαχωρισμός των παραπάνω μεθόδων αναπαράστασης γνώσης σε μεθόδους κατάλληλες για σημασιολογική, δηλωτική και διαδικαστική γνώση αποτυπώνεται στο σχήμα 2.3.



Σχήμα 2.3 Είδη γνώσης και μέθοδοι αναπαράστασής τους

Όλοι οι τρόποι αναπαράστασης της γνώσης αποτελούν μοντέλα θεωριών από το χώρο της Γνωσιακής ή Γνωστικής Επιστήμης. Η ΤΝ παρέχει τα εργαλεία για την υλοποίηση, τη μελέτη και την εξέλιξη των μοντέλων αυτών, αλλά τα χρησιμοποιεί επίσης προσαρμόζοντάς τα στις δικές της ανάγκες για το σχεδιασμό ευφυών συστημάτων.

2.2.3 Σχήματα Λογικής Αναπαράστασης

Για την αναπαράσταση γνώσης οι λογικές αναπαραστάσεις χρησιμοποιούν εκφράσεις της τυπικής λογικής, η οποία είναι μια γλώσσα αναπαράστασης γνώσης που βασίζεται σε λογικά προτασιακά σχήματα και μπορεί να κατηγοριοποιηθεί στα εξής:

- προτασιακός λογισμός,
- κατηγορηματικός λογισμός,
- κατηγορηματικός λογισμός πρώτης τάξης.

Προτασιακός Λογισμός

Στην **προτασιακή λογική** ή **προτασιακό λογισμό** (propositional calculus) ενδιαφερόμαστε για **δηλωτικές προτάσεις** (declarative sentences) που μπορούν να είναι αληθείς (true) ή ψευδείς (false), αλλά όχι και τα δύο.

Παραδείγματα προτάσεων που εκφράζουν δηλώσεις:

Έχει ήλιο.
Κάνει ζέστη.
Θα διψάσεις.

Η αλήθεια ή το ψεύδος που αποδίδεται σε μία πρόταση καλείται **τιμή αλήθειας** (truth value) της πρότασης.

Από απλές δηλωτικές προτάσεις μπορούμε να δομήσουμε σύνθετες προτάσεις (compound propositions), χρησιμοποιώντας τους 5 λογικούς συνδέσμους (logical connectives):

\neg όχι (not): άρνηση
 \wedge και (and): σύζευξη
 \vee ή (or): διάζευξη

→ αν ... τότε (if ... then): συνεπαγωγή
 ↔ αν και μόνο αν (if and only if): ισοδυναμία

Παραδείγματα σύνθετων δηλωτικών προτάσεων είναι:

Έχει ήλιο και κάνει ζέστη. (δήλωση)
 Αν κάνει ζέστη, τότε θα διαβάσεις. (συνεπαγωγή)

Λογικοί σύνδεσμοι στις παραπάνω δύο σύνθετες προτάσεις είναι τα «και» και «εάν ... τότε».

Στην προτασιακή λογική χρησιμοποιούνται σύμβολα (π.χ., P, Q, R) που ονομάζονται **άτομα** (atoms) και παριστάνουν δηλώσεις υπό μορφή προτάσεων είτε αληθείς είτε ψευδείς σε δεδομένη κατάσταση του περιβάλλοντος κόσμου, π.χ.:

P: Έχει ήλιο.
 Q: Κάνει ζέστη.
 R: Θα διαβάσεις.

Οι **καλοσηματισμένοι τύποι** (well-formed formulas) στην προτασιακή λογική είναι είτε απλά άτομα είτε σύνθετοι τύποι που προκύπτουν από το συνδυασμό απλών ατόμων μέσω των λογικών συνδέσμων:

$P \rightarrow Q$ (Αν έχει ήλιο, τότε κάνει ζέστη.)
 $P \vee Q \rightarrow R$ (Αν έχει ήλιο (P) και κάνει ζέστη (Q), τότε θα διαβάσεις (R).)

Η τιμή αλήθειας ενός καλοσηματισμένου τύπου προκύπτει από τις επιμέρους τιμές αλήθειας των ατόμων που συμμετέχουν σ' αυτόν σύμφωνα με τον παρακάτω πίνακα αληθείας:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Πίνακας 2.1 Πίνακας Αληθείας

Κατηγορηματικός Λογισμός

Ο **κατηγορηματικός λογισμός** (predicate calculus) διαθέτει, επιπλέον των ανωτέρω, την ικανότητα να καθορίζει σχέσεις και να κάνει γενικεύσεις σχετικά με τις προτάσεις. Οι λογικές εκφράσεις χρησιμοποιούν τον κατηγορηματικό λογισμό για την εξαγωγή συμπερασμάτων εξετάζοντας το αληθές ή ψευδές των προτασιακών δηλώσεων.

Για την αναπαράσταση γνώσεων στον κατηγορηματικό λογισμό χρησιμοποιούμε:

- **σύμβολα** (symbols) που ονομάζονται *σταθερές* (constants) και παριστάνουν *αντικείμενα* (objects) του περιβάλλοντος κόσμου: π.χ. Maria.
- τα λεγόμενα **συναρτησιακά σύμβολα** (functional symbols) για την κατασκευή δομών που επίσης παριστάνουν αντικείμενα του περιβάλλοντος κόσμου, αλλά που προσδιορίζονται

μέσω άλλων αντικειμένων του ίδιου κόσμου, π.χ. `father_of(MARIA), date(10, June, 1999)`.

- **κατηγορήματα** (predicates), που περιγράφουν ιδιότητες και σχέσεις που αφορούν αντικείμενα τα οποία αντιπροσωπεύονται από σύμβολα.

Π.χ. το κατηγορήμα 'on' μπορεί να χρησιμοποιηθεί, για να εκφραστεί ότι ένα αντικείμενο είναι επάνω σε ένα άλλο δηλαδή, η πρόταση ότι το βάζο είναι πάνω στο τραπέζι μπορεί να γραφεί στον κατηγορηματικό λογισμό ως `on(βάζο, τραπέζι)`.

Τα κατηγορήματα συνδυάζονται και με παραμέτρους, για να εκφραστούν σχέσεις των συμβόλων και παραμέτρων μέσα στις προτάσεις, π.χ. αν δηλωθεί `on(x, τραπέζι)` σημαίνει ότι η παράμετρος x πρέπει να ταυτιστεί με κάποιο από τα αντικείμενα που βρίσκονται πάνω στο τραπέζι.

Αν το κατηγορήμα 'father' χρησιμοποιείται, για να δηλώσει το όνομα του πατέρα ενός ατόμου, π.χ. `father(Maria, Jim)`, τότε από τις κατηγορηματικές προτάσεις `father(Maria, x)` και `father(George, x)` προκύπτει ο λογικός συνειρμός ότι η Ann και ο George έχουν τον ίδιο πατέρα, χωρίς να μας ενδιαφέρει το όνομά του.

- **καλοσχηματισμένους τύπους** (well-formed formulas).

Οι καλοσχηματισμένοι τύποι στην κατηγορηματική λογική, όπως και στην προτασιακή λογική, είναι είτε απλά άτομα είτε σύνθετοι τύποι που προκύπτουν από το συνδυασμό απλών ατόμων μέσω των λογικών συνδέσμων (logical connectives): \neg (άρνηση), \wedge (λογικό ΚΑΙ), \vee (λογικό Ή), \rightarrow (συνεπαγωγή), \leftrightarrow (ισοδυναμία), \in (ανήκει), \forall (δια κάθε).

Π.χ.

Κάθε αντικείμενο πάνω στο τραπέζι είναι κόκκινο:

$\forall x: on(x, table) \rightarrow red(x)$

Όλα τα μανιτάρια είναι ή μωβ ή δηλητηριώδη:

$\forall x: mushroom(x) \rightarrow (purple(x) \vee poisonous(x))$

Κανένα μωβ μανιτάρι δεν είναι δηλητηριώδες:

$\forall x: (mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x)$

Κατηγορηματικός Λογισμός Πρώτης Τάξης

Ο κατηγορηματικός λογισμός βασίζεται στην αλήθεια των προτάσεων και στους συμπερασματικούς μηχανισμούς που ακολουθούν λογικούς συνειρμούς. Ως επέκταση, με την προσθήκη συναρτήσεων σε αυτόν δημιουργείται ο **κατηγορηματικός λογισμός πρώτης τάξης** (first-order predicate calculus). Μια *συνάρτηση* είναι μία λογική κατασκευή που παράγει μια αξία.

Για παράδειγμα, βάσει της δήλωσης ποιος είναι γονέας των x και y μπορούμε να συναγάγουμε μέσω συναρτήσεων διάφορα συμπεράσματα μεταξύ των οποίων αν οι x και y είναι αδέρφια. Κάτι τέτοιο θα γραφόταν ως ακολούθως σε περιβάλλον PROLOG, όπου οι μεταβλητές πρέπει να γράφονται με κεφαλαίους χαρακτήρες:

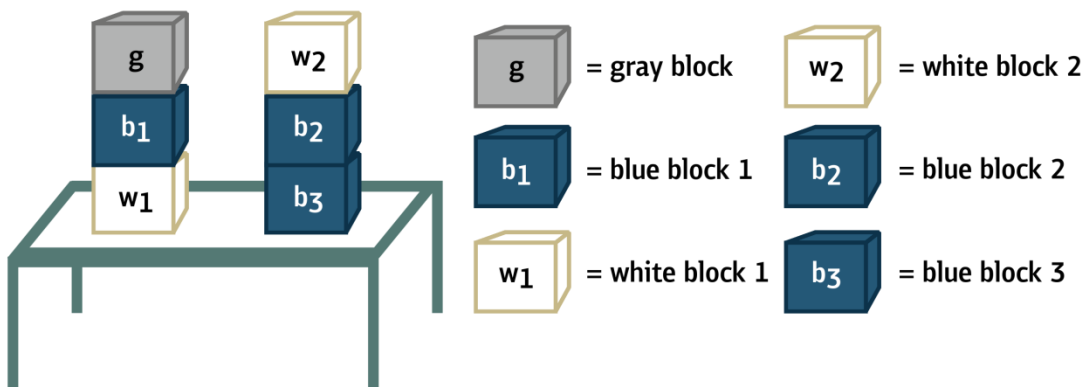
```
 $\forall X,Z,W: father(X,W) \wedge father(Z,W) \rightarrow brothers(X,W):$   
father(john, peter)  
father(mary, peter)  
father(ann, takis)  
? brothers(john, mary)  
yes  
? brothers(mary, ann)  
no
```

Τα προτερήματα αυτού του είδους της αναπαράστασης είναι η απλότητα της σημειολογίας της, το ότι επιτρέπει τόσο την άμεση αντίληψη αυτών που αναπαριστά όσο και την προσθήκη νέων δηλώσεων, χωρίς να επηρεάζονται οι ήδη καταχωρισμένες.

Ο κατηγορηματικός λογισμός είναι κατάλληλος τρόπος αναπαράστασης για χώρους ολοκληρωμένων θεωριών, όπου μπορούν να εφαρμοστούν τεχνικές απόδειξης θεωρημάτων, ώστε να προκύψουν νέα θεωρήματα από παλαιά. Αντίθετα, είναι ακατάλληλος για αναπαράσταση διαδικαστικής και ευρετικής γνώσης, όπως, επίσης, για το χειρισμό μεγάλων βάσεων δεδομένων. Η πιο γνωστή γλώσσα που υλοποιεί τον κατηγορηματικό λογισμό είναι η PROLOG .

Παράδειγμα αναπαράστασης του προβλήματος των κύβων υλοποιημένη σε PROLOG:

Έστω ότι στο πρόβλημα υπάρχουν 6 κύβοι, ένας γκρι (g), τρεις μπλε (b) και 2 άσπροι (w), τοποθετημένοι σε δύο στήλες πάνω σε ένα τραπέζι, όπως στο σχήμα που ακολουθεί:



Σχήμα 2.4 Γραφική αναπαράσταση του προβλήματος των 6 κύβων

Η αναπαράσταση του κόσμου του προβλήματος στο συγκεκριμένο στιγμιότυπο, σε PROLOG, θα έχει ως ακολούθως:

Δηλώσεις για τις σχετικές θέσεις των 6 κύβων:

```
isabove(g, b1)
isabove(b1, w1)
isabove(w2, b2)
isabove(b2, b3)
```

Δηλώσεις για τα χρώματα των 6 κύβων:

```
color(g, gray)
color(b1, blue)
color(b2, blue)
color(b3, blue)
color(w1, white)
color(w2, white)
```

Καλοσηματισμένος τύπος που περιγράφει πότε ένας κύβος είναι επάνω από κάποιον άλλον σε μια στήλη κύβων:

`isabove(X, Z) if isabove(X, Y) and isabove(Y, Z)`

Ερωτήσεις που μπορούν να τεθούν και αντίστοιχες λογικές απαντήσεις:

```
?isabove(b2, w1)
no
?color(w1, X)
X=white
?color(X, blue)
X=b1, X=b2, X=b3
```

Δηλαδή, η μεταβλητή X μπορεί να δεσμευτεί με εναλλακτικά με τις τιμές b1, b2, b3 που αντιστοιχούν στα ονόματα των τριών μαύρων κύβων.

2.2.4 Δομημένες Μορφές Αναπαράστασης Γνώσης

Οι λογικές αναπαραστάσεις δεν μπορούν εύκολα να αναπαραστήσουν κλάσεις αντικειμένων και είναι πολύ αυστηρές στο φορμαλισμό τους. Αντίθετα, οι δομημένες μορφές αναπαράστασης δίνουν αρκετή ελευθερία στον τρόπο αναπαράστασης, επιτρέποντας με αυτόν τον τρόπο μία περισσότερο διαισθητική προσέγγιση απ' όσο οι λογικές.

Δομημένες μορφές αναπαράστασης είναι τα σημασιολογικά δίκτυα, τα σχήματα, τα πλαίσια και τα σενάρια.

Όλες οι δομημένες μορφές αναπαράστασης περιλαμβάνουν:

- Έννοιες
- Διάφορα είδη των δεσμών μεταξύ των εννοιών
 - "έχει-μέρος" (has-part ή aggregation),
 - "ανήκει" (is-a ή specialization),
 - πιο εξειδικευμένες σχέσεις ανάλογα με τον τομέα.
- Συνήθως περιλαμβάνουν, επίσης,
 - κληρονόμηση (inheritance),
 - κάποιο είδος διαδικαστικού συνημμένου.

Περισσότερα για τα παραπάνω θα αναφερθούν στη συνέχεια.

Σημασιολογικά Δίκτυα

Πρώτοι οι Collins & Quillian (1969, 1970) ανέπτυξαν τα σημασιολογικά δίκτυα ως υπολογιστικά μοντέλα αναπαράστασης της γνώσης στο πλαίσιο των πρώτων ερευνών στο χώρο της Γνωστικής Επιστήμης.

Με τα **σημασιολογικά δίκτυα** (semantic networks) αποτυπώνεται γραφικά η υποκείμενη οργάνωση των ιδεών για έναν τομέα του επιστητού (π.χ. ποδόσφαιρο, Κοινωνιολογία).

Ένα σημασιολογικό δίκτυο είναι μία προσέγγιση των ιδιοτήτων και των σχέσεων των αντικειμένων, γεγονότων, στοιχείων, καταστάσεων και ενεργειών που το αποτελούν. Οι σχέσεις μπορεί να είναι γενικού ή ειδικού τύπου. Χαρακτηριστικό τους είναι η δυνατότητα που έχουν να υποστηρίζουν κληρονόμηση ιδιοτήτων, η οποία με τη σειρά της επιτρέπει την εξαγωγή συμπερασμάτων.

Σε ένα σημασιολογικό δίκτυο υπάρχουν κόμβοι, συνδέσεις και χαρακτηρισμοί συνδέσεων. Κάθε σύνδεση συνδέει ένα κόμβο-ουρά με έναν κόμβο-κεφαλή. Οι κόμβοι και οι συνδέσεις υποδεικνύουν οντότητες που αφορούν ένα συγκεκριμένο κόσμο προβλήματος.

Τα σημασιολογικά δίκτυα είναι ένα είδος αναπαράστασης που διαθέτει ιδιότητες για κατασκευή ενός κόμβου ή μιας νέας σύνδεσης δεδομένων δύο κόμβων, για απόδοση χαρακτηρισμού σε μια νέα σύνδεση και για δημιουργία: :

- μιας λίστας όλων των συνδέσεων που ξεκινούν από/καταλήγουν σε ένα δεδομένο κόμβο,
- ενός κόμβου “ουράς” ή “κεφαλής” δεδομένης μίας σύνδεσης,
- ενός χαρακτηρισμού σύνδεσης, δεδομένης της σύνδεσης.

Κάθε κόμβος ενός σημασιολογικού δικτύου θεωρείται ως στοιχείο και μπορεί να διακριθεί σε τέσσερα είδη:

- έννοιες (concepts), οι οντότητες (τα αντικείμενα) που θέλουμε να απεικονίσουμε στο δίκτυο,
- περιστατικά (events), τα γεγονότα που λαμβάνουν χώρα,
- χαρακτηριστικά (characteristics), οι ιδιότητες των οντοτήτων,
- αξίες (values), οι τιμές ιδιοτήτων.

Τα σημασιολογικά δίκτυα έχουν τη μορφή δένδρου με τα εξής χαρακτηριστικά:

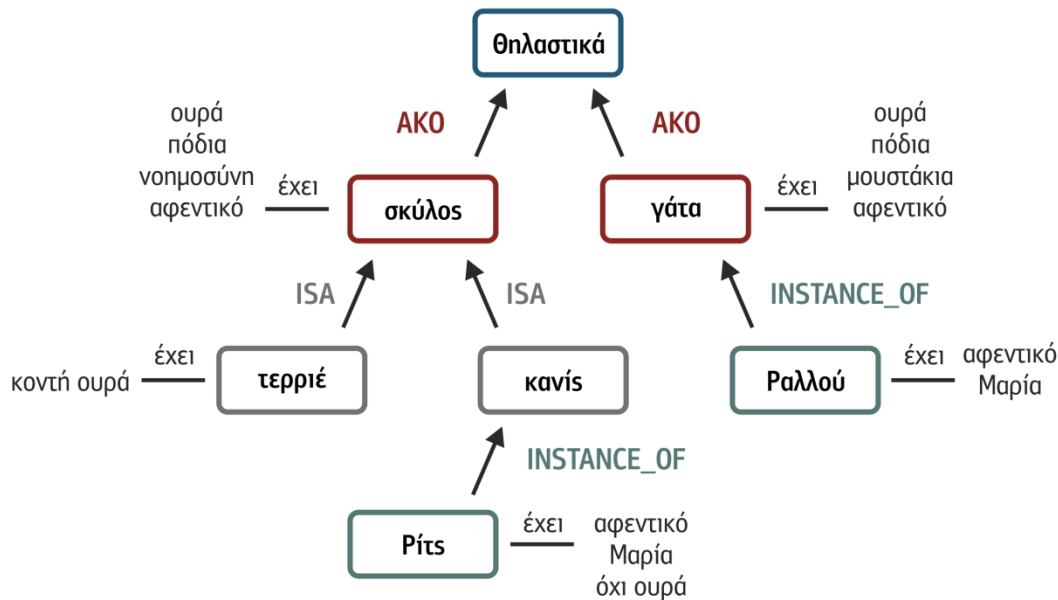
- Οι συνδέσεις μεταξύ των κόμβων καλούνται *κλάδοι (branches)*.
- Κάθε κλάδος συνδέει δυο κόμβους, ο κόμβος-κεφαλή καλείται *κόμβος-γονέας (parent node)* και ο δεύτερος κόμβος καλείται *κόμβος-παιδί (child node)*.
- Ο πρώτος κόμβος του δικτύου δεν έχει γονέα και καλείται *κόμβος-ρίζα (root node)*.
- Όσοι κόμβοι δεν έχουν παιδιά καλούνται *φύλλα (leaves)*.
- Όταν δύο κόμβοι συνδέονται με μια αλυσίδα από δύο οι περισσότερους κόμβους, ο ένας καλείται *πρόγονος (ascendant)* και ο άλλος *απόγονος (descendant)*.
- Οι συνδέσεις του σημασιολογικού δικτύου επιτρέπουν σε έναν κόμβο-παιδί να κληρονομήσει τις ιδιότητες και τα χαρακτηριστικά του κόμβου-πατέρα.

Οι κόμβοι-παιδιά μπορούν να οργανωθούν σε κλάσεις και οι δε κλάσεις σε ιεραρχίες με σχέσεις της μορφής:

- *a_kind_of* (δεσμός AKO)
Η σχέση **AKO** υπάρχει μεταξύ κόμβων οντοτήτων που αντιπροσωπεύουν κλάσεις αντικειμένων. Σε κόμβο που συνδέεται με κάποιον άλλον με σχέση AKO μπορούν να προστεθούν νέοι δεσμοί που προσδίδουν νέες ιδιότητες.
- *is_a* (δεσμός ISA)
Η σχέση **ISA** είναι παρόμοια με τη σχέση AKO, με τη διαφορά ότι ο συνδεόμενος κόμβος δεν αντιπροσωπεύει κλάση αλλά υποκλάση οντοτήτων και, ως εκ τούτου, δεν επιτρέπεται να του προστεθούν νέες ιδιότητες, παρά μόνον να κληρονομηθούν οι ήδη υπάρχουσες ιδιότητες από κόμβους κλάσεων ψηλότερα στην ιεραρχία ή οι ιδιότητες που κληρονομούνται να αλλάξουν τιμές.
- *instance_of* (δεσμός INSTANCE_OF)
Η σχέση *instance_of* υπάρχει μόνο μεταξύ κόμβων οντοτήτων που αποτελούν στιγμιότυπα του κόσμου που απεικονίζεται στη δομή και κόμβων γενικότερων κλάσεων.

Ένα σημασιολογικό δίκτυο μπορεί να είναι ιεραρχικό (βλέπε Σχήμα 2.5) ή ετεροαρχικό (βλέπε Σχήμα 2.6).

Τα **ιεραρχικά σημασιολογικά δίκτυα** (hierarchical semantic networks) έχουν όλα τα χαρακτηριστικά των σημασιολογικών δικτύων μόνο που όλοι οι άλλοι κόμβοι έχουν ακριβώς έναν γονέα.

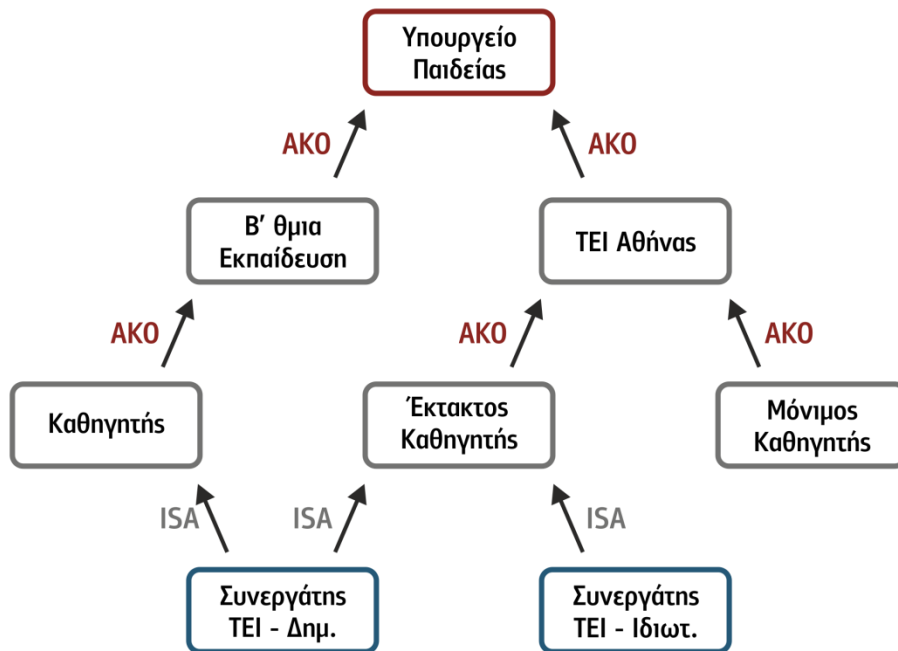


Σχήμα 2.5 Παράδειγμα ιεραρχικού σημασιολογικού δικτύου

Στο ιεραρχικό σημασιολογικό δίκτυο του σχήματος 2.5,

- οι οντότητες είναι οι διάφορες μορφές ζωής (π.χ. θηλαστικά, τίγρης, Τζίμις).
- περιστατικά δεν υπάρχουν.
- ιδιότητες είναι τα παράγει, καταναλώνει, γεννά, έχει, χρώμα, αρέσει, ηλικία και ιδιοκτήτης με τις αντίστοιχες τιμές τους.

Η σχέση AKO μεταξύ Μορφής Ζωής και Θηλαστικού δηλώνει ότι τα θηλαστικά ανήκουν μεν στις μορφές ζωής και κληρονομούν τα χαρακτηριστικά τους, αλλά είναι κλάση που διαθέτει και δικά της χαρακτηριστικά, όπως ότι γεννά ζωντανά παιδιά, που τη διαφοροποιούν από άλλες κλάσεις που ανήκουν στις μορφές ζωής, όπως για παράδειγμα τα πτηνά. Κάτι ανάλογο συμβαίνει και μεταξύ θηλαστικών και τίγρεων. Με σχέση ISA ή is_a συνδέονται οι ινδικές τίγρεις και οι τίγρεις, διότι οι ινδικές τίγρεις αποτελούν υπο-τάξη της μεγαλύτερης τάξης των τίγρεων και, ως εκ τούτου, κληρονομούν όλα τα χαρακτηριστικά τους, χωρίς να διαθέτουν άλλα επιπλέον και μόνο τιμές αυτών των χαρακτηριστικών μπορούν να διαφοροποιήσουν, όπως για παράδειγμα την τιμή του χαρακτηριστικού χρώματος που έχει τιμή καφέ αντί της τιμής κίτρινο που είχε η κλάση των τίγρεων.



Σχήμα 2.6 Παράδειγμα ιεραρχικού σημασιολογικού δικτύου

Τα **ετεροαρχικά σημασιολογικά δίκτυα** (non-hierarchical neural network) έχουν όλα τα χαρακτηριστικά των ιεραρχικών σημασιολογικών δικτύων, με τη μόνη διαφορά ότι υπάρχει τουλάχιστον ένας ενδιάμεσος κόμβος που έχει παραπάνω από ένα γονέα.

Το σημασιολογικό δίκτυο του σχήματος 2.6 αναπαριστά γνώση για τους καθηγητές του TEI Αθήνας,

- Οι καθηγητές μπορεί να ανήκουν στην κλάση Μόνιμος Καθηγητής ή στην κλάση Έκτακτος Καθηγητής.
- Οι έκτακτοι καθηγητές διαχωρίζονται σε καθηγητές της Β/θμιας εκπαίδευσης (υποκλάση Συεργάτης TEI-Δημ.) ή ιδιώτες (υποκλάση Συεργάτης TEI-Ιδιωτ.).
- Στην περίπτωση της υποκλάσης Συεργάτης TEI-Δημ., αυτή ανήκει με σχέση ISA και στους καθηγητές που αποτελούν κλάση της Β/θμιας Εκπαίδευσης και στους έκτακτους καθηγητές που αποτελούν κλάση του TEI Αθήνας.

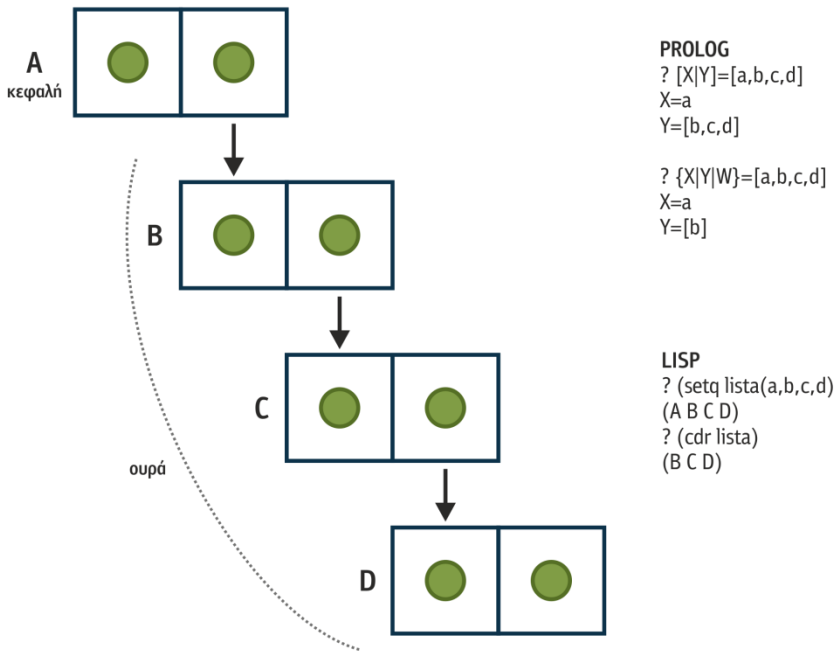
Τα σημασιολογικά δίκτυα αποτυπώνονται εύκολα μέσα σε δομές συνδεδεμένων λιστών στον Η/Υ, με κόμβους που αποτελούνται από δύο μέρη: το πρώτο περιέχει το όνομα και τις ιδιότητες της οντότητας που απεικονίζει ο κόμβος και το δεύτερο τη διεύθυνση του επόμενου λογικά κόμβου. Το δεδομένο του πρώτου κόμβου καλείται κεφαλή, ενώ τα δεδομένα όλων των υπόλοιπων κόμβων καλούνται ουρά. Ο δείκτης του τελευταίου κόμβου της αλυσίδας που σχηματίζεται περιέχει ένα ειδικό σύμβολο ένδειξης τέλους, που συνήθως ονομάζεται κενός δείκτης (null pointer ή NIL).

Το παρακάτω παράδειγμα παρουσιάζει την υλοποίηση σε λίστα συνάφειας του ιεραρχικού σημασιολογικού δικτύου του αποτυπώνεται στο [σχήμα 2.5](#):

```
( (θηλαστικά ())
  (σκύλος (είδος θηλαστικό) (ουρά NAI) (πόδια NAI) (νοημοσύνη NAI)
    (αφεντικό NAI))
  (γάτα (είδος θηλαστικό) (ουρά NAI) (πόδια NAI) (μουστάκια NAI)
    (αφεντικό NAI))
  (τερριέ (είναι σκύλος) (ουρά KONTH))
  (κανίς (είναι σκύλος) )
  (Ριτς (στιγμιότυπο κανίς) (ουρά OXI) (αφεντικό Κώστας) )
  (Ραλλού (στιγμιότυπο γάτα) (αφεντικό Μαρία) )
```

)

Στο σχήμα 2.7 παρουσιάζεται η γενικότερη δομή συνδεδεμένης λίστας, στην οποία μπορεί να αποτυπωθεί ένα σημασιολογικό δίκτυο και οι τρόποι επεξεργασίας της από διάφορες γλώσσες TN, όπως PROLOG και LISP.



Σχήμα 2.7 Παράδειγμα υλοποίησης και επεξεργασίας σημασιολογικού δικτύου σε μορφή συνδεδεμένης λίστας

Παρακάτω παρουσιάζεται η αποτύπωση του ιεραρχικού σημασιολογικού δικτύου του [σχήματος 2.5](#) υπό μορφή συνδεδεμένης λίστας σε γλώσσα Lisp.

```
(defun *mammal* ()
'((dog (kind_of mammal)
      (tail YES)
      (feet YES)
      (intelligence YES)
      (boss YES))
  (cat ((kind_of mammal)
      (tail YES)
      (feet YES)
      (moustaches YES)
      (boss YES))
  (terrier (is-a dog)
          (tail short))
  (canis (is-a dog))
  (Rits (instance canis)
        (tail no)
        (boss Costas))
  (Rallou (instance cat)
          (boss Maria))))
```

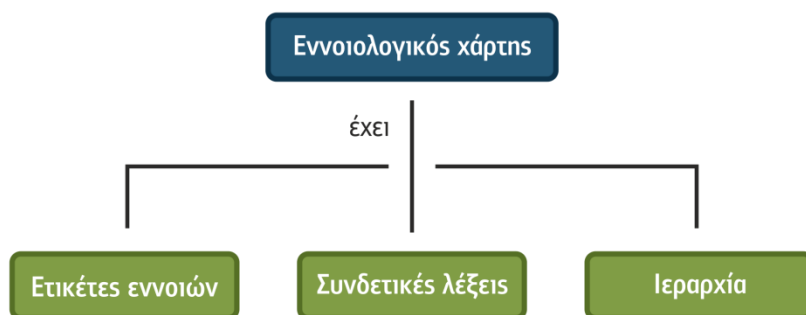
Για μια τέτοια αναπαράσταση σε Lisp χρησιμοποιείται η συνάρτηση assoc για να εντοπιστεί μια οντότητα.

Τα προτερήματα των σημασιολογικών δικτύων είναι η ευχέρειά τους στην προσθήκη, αλλαγή και διαγραφή κόμβων, η ικανότητα των κόμβων να κληρονομούν χαρακτηριστικά και ιδιότητες από άλλους κόμβους και η ευκολία με την οποία συνάγονται συμπεράσματα σχετικά με την κληρονομούμενη ιεραρχία. Το μειονέκτημά τους είναι ότι μόνο απλές άλλες αναπαραστάσεις, όπως προτασιακές και τα πλαίσια που παρουσιάζονται παρακάτω, μπορούν να προκύψουν από αυτά.

Τα σημασιολογικά δίκτυα ως εννοιολογικοί χάρτες

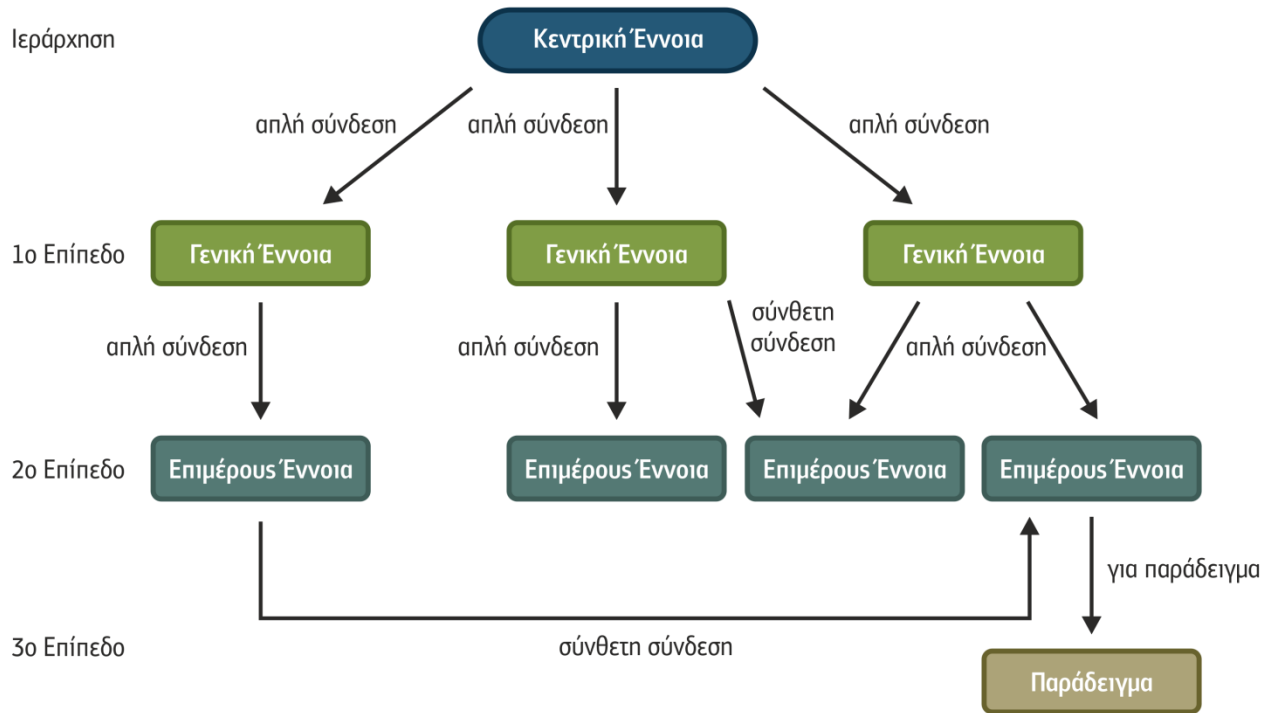
Καρπός της προσπάθειας να λειτουργήσουν τα σημασιολογικά δίκτυα ως γνωστικά εργαλεία που εμπλέκουν τους μαθητές στην ανάλυση των διαρθρωτικών σχέσεων εντός του υπό μελέτη γνωστικού περιεχομένου είναι οι εννοιολογικοί χάρτες. . Αν παραδεχτούμε ότι ουσιαστικά η μνήμη μας είναι ένα σημασιολογικό δίκτυο, με τις γνώσεις μας να αποθηκεύονται στους νευρώνες του εγκεφάλου μας, τότε η μάθηση μπορεί να θεωρηθεί ως μια αναδιοργάνωση της σημασιολογικής μας μνήμης. Η παραγωγή νέων εννοιών ή ιδιοτήτων τους μέσα σε ένα σημασιολογικό δίκτυο αντανακλά αλλαγές που έχουν πραγματοποιηθεί στη σημασιολογική μνήμη, δεδομένου ότι τα σημασιολογικά δίκτυα περιγράφουν την υπάρχουσα γνώση. Με τον τρόπο αυτό, προγράμματα σημασιολογικής δικτύωσης μπορούν να χρησιμοποιηθούν για να απεικονίσουν την απόκτηση γνώσεων.

Ένας **εννοιολογικός χάρτης- EX** (concept map) συνδυάζει την έκφραση των εννοιών (με μορφή εικόνων, σχημάτων ή λέξεων) με τη σύνδεση αυτών ως κόμβων, προσδιορίζοντας την ιεραρχία μεταξύ τους.



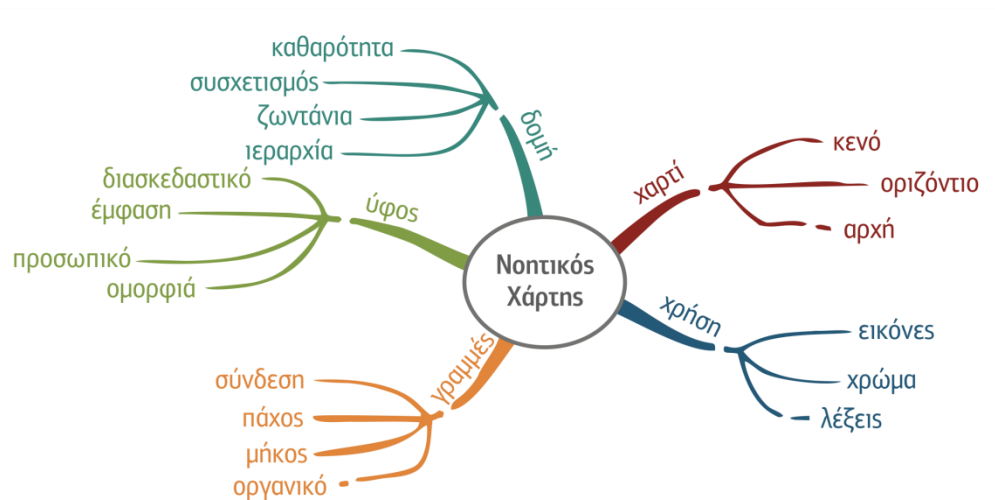
Σχήμα 2.8 Τμήματα από τα οποία δομείται ένας εννοιολογικός χάρτης

Η σύνδεση γίνεται μεταξύ κόμβων οι οποίοι αφορούν διαφορετικές ομάδες εννοιών (κλάσεις) που ανήκουν σε διάφορους τομείς με τρόπο παρόμοιο με αυτόν με τον οποίο σχεδιάζεται ένα περίγραμμα, αλλά λιγότερο γραμμικά και περισσότερα χωρικά.



Σχήμα 2.9 Η Δομή ενός Εννοιολογικού Χάρτη

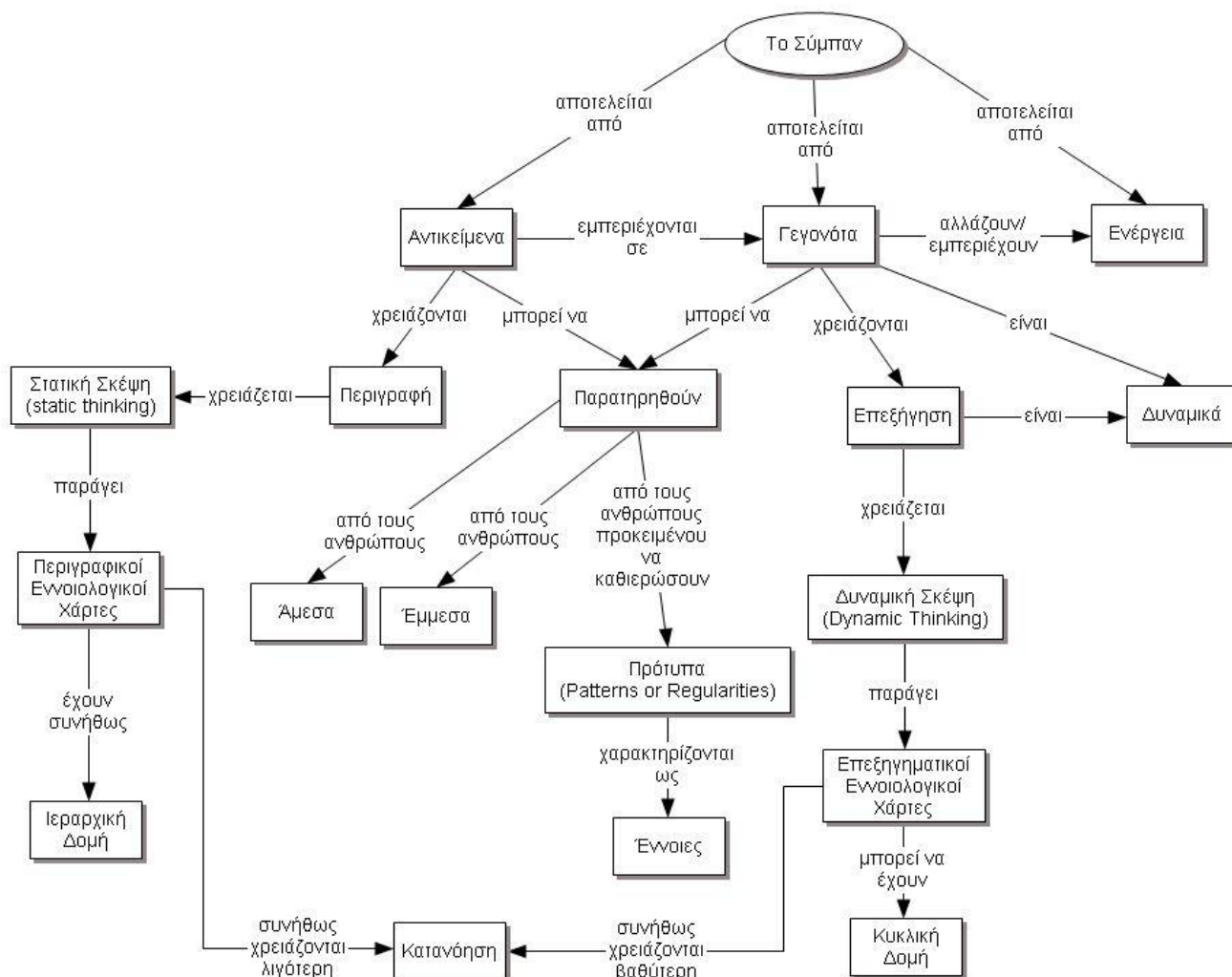
Παρεμφερώς, ο νοητικός χάρτης (mind map) αναφέρεται σε έναν εννοιολογικό χάρτη που καταγράφει γνώση για μία μόνη έννοια. Μπορεί να θεωρηθεί ως ένα διάγραμμα οπτικοποίησης πληροφοριών, στο οποίο μια κεντρική έννοια αναλύεται σε ένα αραχνοειδές διάγραμμα (spider diagram - tree like diagram) και περιμετρικά της κεντρικής έννοιας συνδέονται άλλες προτάσεις που αναλύουν την κεντρική έννοια. Οι σύνδεσμοι σε ένα νοητικό χάρτη είναι «παθητικοί» Δείχνουν απλώς μια σχέση ιεραρχική ανάμεσα στα δύο άκρα.



Σχήμα 2.10 Παράδειγμα δομής νοητικού χάρτη

Η τεχνική της εννοιολογικής χαρτογράφησης αναπτύχθηκε από τον J. D. Novak (Novak & Gowin, 1984, Novak, 1991) βάσει της θεωρίας του D. Ausubel (1968). Στην πράξη, αντικατοπτρίζεται πώς ο εγκέφαλός μας κάνει πραγματικά συνδέσεις μεταξύ των γνώσεων που προσλαμβάνει. Οι εννοιολογικοί χάρτες μπορούν να

συνδεθούν μεταξύ τους, ώστε να δομήσουν **μοντέλα γνώσης** (knowledge models), όπως το παρακάτω ολοκληρωμένο μοντέλο γνώσης.



Σχήμα 2.11 Ολοκληρωμένο μοντέλο γνώσης αποτυπωμένο ως σύνθετος εννοιολογικός χάρτης⁷

Βήματα κατασκευής ενός εννοιολογικού χάρτη:

1. Επιλέξτε το κείμενο και απαριθμήστε τη λίστα των βασικών εννοιών.
2. Οργανώστε τις έννοιες (ιεραρχία).
3. Κατασκευάστε χάρτη χρησιμοποιώντας μία αξιολογική σειρά για τις έννοιες.
4. Συμπεριλάβετε έγκυρες συνδέσεις (προτάσεις), διασυνδέσεις μεταξύ εννοιών και παραδείγματα (γεγονότα ή αντικείμενα).
5. Αξιολογήστε.

Στο πλαίσιο ενός εκπαιδευτικού σχεδιασμού ένας εννοιολογικός χάρτης μπορεί να αξιοποιηθεί ως:

- μέσο οργάνωσης και παρουσίασης του μαθήματος,
- διαγνωστικό εργαλείο για την ανίχνευση των πρότερων γνώσεων των μαθητών,

⁷ Ο χάρτης έχει δημιουργηθεί στα Ελληνικά με βάση αντίστοιχο χάρτη που υπάρχει στη δημοσίευση των Cañas και Novak (2006).

- εργαλείο
 - επίλυσης προβλημάτων,
 - συνεργασίας,
 - ανάπτυξης και καλλιέργειας κριτικής και δημιουργικής σκέψης,
 - αξιολόγησης της μάθησης.

Τα παραπάνω μπορούν να υλοποιηθούν μέσω εκπαιδευτικών δραστηριοτήτων, όπως:

- συμπλήρωση EX με έννοιες και συνδέσεις που λείπουν,
- επέκταση EX με προσθήκη εννοιών και συνδέσεων,
- διόρθωση EX,
- σχολιασμός EX είτε ελεύθερα είτε με απάντηση σε ερωτήσεις,
- κατασκευή EX στον οποίο δίνονται:
 - η κεντρική έννοια,
 - ένα σύνολο εννοιών,
 - ένα σύνολο εννοιών και συνδέσεων,
 - μια ερώτηση,
 - ένα σχετικό κείμενο,
 - συνδυασμός των παραπάνω.

Σχήματα

Ένα **σχήμα** (schema) είναι μία δομημένη κλάση εννοιών που περιέχει συνήθως γενική γνώση και η οποία μπορεί να χρησιμοποιηθεί για να αναπαραστήσει γεγονότα, σειρά γεγονότων, καταστάσεις, σχέσεις και αντικείμενα.

Τα σχήματα ήταν μία από τις πρώτες προτάσεις των ερευνητών στη προσπάθειά τους να δώσουν μία ερμηνεία για την οργάνωση των σύνθετων πληροφοριών στη μνήμη του ανθρώπου, με τρόπο που να ερμηνεύει την ανθρώπινη συμπεριφορά.

Η θεωρία περί σχημάτων απορρέει από μία σειρά φιλοσοφικών θεωριών που ξεκινούν από αυτές του E. Kant (1724-1804), συνεχίζουν με αυτές του J. Piaget (1896-1980) και καταλήγουν με πιο σύγχρονες προτάσεις στα τέλη της δεκαετίας του 1970 και στις αρχές της δεκαετίας του 1980 (Rumelhart & Ortony 1977, Rumelhart, 1980), των οποίων το οντολογικό επιχείρημα είναι ότι υπάρχουν μοριακά φαινόμενα (π.χ. εκδηλώσεις, σκηνές σε χώρο, δομή λόγου) στον (ψυχολογικό) κόσμο, και ότι τα σχήματα είναι οι κατάλληλες μορφές αναπαράστασης γνώσης γι' αυτά τα φαινόμενα. Παρόμοιες θεωρίες που αναφέρονται ως «θεωρίες σεναρίων ή ιστοριών» παρουσιάζονται από πολλούς ερευνητές από τους οποίους οι πιο γνωστοί είναι οι R. Schank και R.P. Abelson (1977), καθώς και ο D.E. Rumelhart (1975). Στο χώρο της TN ο Marvin Minsky (1975) προτείνει παρόμοιες δομές που καλούνται **πλαίσια** και μέχρι σήμερα χρησιμοποιούνται ευρέως για αναπαράσταση της γνώσης, κυρίως στα έμπειρα συστήματα.

Πλαίσια

Τα **πλαίσια** (frames) αποτυπώνουν κυρίως δηλωτική γνώση και είναι δύο κατηγοριών:

- **πλαίσια κλάσεων** ή **κλάσεις** (class): περιγράφουν μία ανώτερη κλάση πραγμάτων (π.χ. πλαίσιο φρούτο)
- **πλαίσια στιγμιοτύπων** (instance frames): περιγράφουν ένα συγκεκριμένο στοιχείο (π.χ. παρακάτω το πλαίσιο MyCourse).

Ως προς τη δομή τους, τα πλαίσια είναι πιο κοντά από τα άλλα είδη αναπαράστασης στην έννοια της εγγραφής (record) στην πληροφορική.

My Course	
Slot Name	Filler
is_a	AI course
Date	Spring 2015
Time	Monday 2pm
Instructor	Smith
Enrollment	80
Has	Laboratory

Πίνακας 2.2 Παράδειγμα πλαισίου στιγμιότυπου

Τα πλαίσια αποτελούνται από **σχισμές** (slots) που περιέχουν έννοιες ή παραπέμπουν σε άλλα πλαίσια.

Οι σχισμές μπορούν να περιέχουν όλα τα είδη των στοιχείων:

- κανόνες, γεγονότα, φωτογραφίες, βίντεο, σχόλια, πληροφορίες από εντοπισμό σφαλμάτων, ερωτήσεις, υποθέσεις, άλλα πλαίσια,
- διαδικαστικά συνημμένα, όπως διαδικασίες στις οποίες γίνεται επίκληση σε ειδικές καταστάσεις (π.χ. στην υλοποίηση των πλαισίων στον αντικειμενοστραφή προγραμματισμό με τη χρήση daemons),
- οδηγίες σχετικά με τη δημιουργία, την τροποποίηση ή την αφαίρεση της αξίας μια σχισμής.

Για παράδειγμα, η γενική δηλωτική γνώση που έχουμε για τα φρούτα θα μπορούσε να αναπαρασταθεί ιεραρχικά με πλαίσια ως εξής:

Πλαίσιο κλάσης: φρούτο (κλάση)

Σχισμή 1: φαγώσιμο (default yes)

Σχισμή 2: βασική διατροφή (default yes)

Σχισμή 3: εύπεπτο (default no)

Πλαίσιο κλάσης: μήλο (κλάση)

ΕΙΔΟΣ: φρούτο

Σχισμή εποχή: χειμερινό

Σχισμή γεύση: εύγευστο

Πλαίσιο στιγμιότυπο: GOLDEN (στιγμιότυπο)

ΕΙΝΑΙ: μήλο

Σχισμή χρώμα: κίτρινο

Σχισμή οσμή: αρωματική

Σχισμή εύπεπτο: yes

Στην παραπάνω αναπαράσταση,

- η ειδική θέση ΕΙΔΟΣ παραπέμπει σε πλαίσιο υψηλότερης τάξης, δημιουργώντας έτσι συνδέσεις του τύπου a_kind-of ή ΑΚΟ, οι οποίες καθορίζουν την ιεραρχία των πλαισίων κλάσης,
- η ειδική θέση ΕΙΝΑΙ δημιουργεί συνδέσεις τύπου is_a ή ISA που αφορούν προσαρτήσεις στιγμιότυπων στις κλάσεις στις οποίες ανήκουν.

Τα πλαίσια είναι ένας εύκολος τρόπος να δει κανείς ένα σημασιολογικό δίκτυο σε ανώτερο επίπεδο από εκείνο όπου κάποιος βλέπει μόνο κόμβους και συνδέσεις. Κάθε κόμβος και όλες οι συνδέσεις που ξεκινούν από αυτόν μπορούν να συγκεντρωθούν και να απεικονιστούν ως ένα πλαίσιο. Το όνομα του πλαισίου αντιστοιχεί στο όνομα του κόμβου και τα ονόματα των θέσεων αντιστοιχούν στα ονόματα των συνδέσεων. Οι έννοιες ή αξίες που περιέχονται στις θέσεις αντιστοιχούν στις έννοιες ή αξίες που απεικονίζουν οι κόμβοι στους οποίους καταλήγουν οι συνδέσεις. Για παράδειγμα, το ιεραρχικό σημασιολογικό δίκτυο του [σχήματος 2.5](#) θα μπορούσε να γραφεί με μορφή συστήματος πλαισίων (και με εμπλουτισμένες ιδιότητες στο πλαίσιο θηλαστικά) ως εξής:

Πλαίσιο: θηλαστικά

Σχισμή 1: έμφυχο

Σχισμή 2: γεννά μωρά1

Πλαίσιο: σκύλος

ΕΙΔΟΣ: θηλαστικό

Σχισμή ουρά: ΝΑΙ

Σχισμή πόδια: ΝΑΙ

Σχισμή νοημοσύνη: ΝΑΙ

Σχισμή αφεντικό: ΝΑΙ

Πλαίσιο: γάτα

ΕΙΔΟΣ: θηλαστικό

Σχισμή ουρά: ΝΑΙ

Σχισμή πόδια: ΝΑΙ

Σχισμή μουστάκια: ΝΑΙ

Σχισμή αφεντικό: ΝΑΙ

Πλαίσιο: κανίς

ΕΙΔΟΣ: σκύλος

Πλαίσιο: τερριέ

ΕΙΔΟΣ: σκύλος

Σχισμή ουρά: ΚΟΝΤΗ

Πλαίσιο στιγμιότυπου: Ριτς

ΕΙΝΑΙ: κανίς

Σχισμή ουρά: ΟΧΙ

Σχισμή αφεντικό: ΚΩΣΤΑΣ

Πλαίσιο στιγμιότυπου: Ραλλού

ΕΙΝΑΙ: γάτα

Σχισμή αφεντικό: ΜΑΡΙΑ

Ένα πλαίσιο στιγμιότυπου κληρονομεί όλες τις θέσεις που έχουν οι ανώτερες κλάσεις του και έχει επιπλέον θέσεις που το χαρακτηρίζουν. Ορισμένες κληρονομούμενες θέσεις ανώτερων κλάσεων επιτρέπεται να αλλάζουν τις προκαθορισμένες (*default*) αξίες τους και άλλες όχι. Θα μπορούσε, δηλαδή, η σχισμή πόδια να τοποθετηθεί στο πλαίσιο θηλαστικό με προκαθορισμένη αξία 4, να το κληρονομούν τα πλαίσια γάτα και σκύλος, αλλά σε ένα νέο πλαίσιο άνθρωπος να μπορεί να διαφοροποιηθεί η αξία της θέσης σε 2 και στο πλαίσιο φάλαινα να έχει αξία 0. Τα δεδομένα των θέσεων μπορούν να υπάρχουν κατά τη φάση της δημιουργίας ενός πλαισίου ή να προκύπτουν μετά απ' αυτήν.

Επομένως, ένα σύστημα πλαισίων είναι μία αναπαράσταση αντίστοιχη με ένα σημασιολογικό δίκτυο όπου:

- Η γλώσσα των κόμβων και των συνδέσεων έχει αντικατασταθεί με αυτήν των πλαισίων και θέσεων.
- Οι θέσεις τύπου *είδος* καθορίζουν μία ιεραρχία κλάσεων πλαισίων.
- Οι θέσεις τύπου *είναι* καθορίζουν σε ποιες κλάσεις ανήκουν τα πλαίσια στιγμιότυπων.
- Υπάρχουν προκαθορισμένες αξίες θέσεων οι οποίες έχουν τη δυνατότητα, επίσης, να θέτουν περιορισμούς στις δεκτές τιμές.
- Υπάρχει μία βασική διαδικασία που διευθετεί όλες τις υπόλοιπες διαδικασίες αναφοράς στα πλαίσια, τηρώντας την ιεραρχία των κλάσεων.

Διαθέτει επίσης δυνατότητες, όπως να δημιουργεί :

- ένα νέο πλαίσιο κλάσης, δεδομένης μίας σειράς υπερ-κλάσεων και ενός συνόλου θέσεων,
- ένα πλαίσιο στιγμιότυπου, δεδομένου ενός συνόλου άμεσων υπερ-κλάσεων,
- διαδικασίες που να περιγράφουν τι να γίνει σε κάθε περίπτωση αναφοράς σε ένα πλαίσιο καθιστώντας με τον τρόπο αυτό την αποτυπωμένη γνώση διαδικαστική.

Η αναζήτηση σε ένα σύστημα πλαισίων μπορεί να γίνει με τις κλασικές μεθόδους αναζήτησης σε δίκτυα που αναφέρονται στο [κεφάλαιο 1](#) με μεθόδους βασισμένες σε κανόνες που αναφέρονται στο κεφάλαιο συστημάτων κανόνων.

Παράδειγμα χρήσης πλαισίων στη PROLOG:

Έστω ότι θέλουμε να περιγράψουμε τη γενική κατηγορία φρούτων με όνομα `fruit`, ένα συγκεκριμένο είδος φρούτου το `apple` και τέλος το συγκεκριμένο τύπο φρούτου το `golden`. Τότε θα είχαμε:

```
frame(fruit, [[edible, [default, yes]],
             [nutrition, [default, yes]],
             [digestible, [default, yes]],
             [essential, [value, yes]]])
```

όπου: `fruit` το όνομα του πλαισίου και τα `[edible,[default,yes]]`, `[nutrition,[default.yes]]`, `[digestible,[default,yes]]`, `[essential,[value,yes]]`, αποτελούν σχισμές του πλαισίου `food` όπου το πρώτο λεκτικό είναι το όνομα της σχισμής (π.χ. `edible`) και ακολουθεί η συνήθης τιμή (`default`) ή η τιμή που ισχύει σε κάθε περίπτωση (`value`).

Για να δηλωθούν οι μεταξύ τους σχέσεις, τα πλαίσια στη `Prolog` , συνδέονται με συναρτήσεις του τύπου:

```
frame(apple, [[link, [value, fruit]]])
```

Παράδειγμα χρήσης πλαισίων στη Common LISP:

Για την ίδια κατηγορία φρούτων, όπως παραπάνω, θα είχαμε:

```
(define-frame fruit
  (edible           :default-values ( yes))
  (nutrition        :default-values ( yes))
  (essential        :default-values ( yes))
  (digestible       :default-values ( yes)))
(define-frame apple(:is fruit)
  (season winter))
```

```
(taste tasteful))
(define-instance GOLDEN (:is apple)
  (color yellow)
  (smell nice)
  (season autumn))
```

Παράδειγμα χρήσης αντικειμενοστραφούς περιβάλλοντος στη LISP:

Κλάση με όνομα `fruit` που ανήκει στην κλάση `food`:

```
(defclass food () (.....))
(defclass fruit (food)
  ((color :initarg :color)
   (price :initarg :price)))
(defclass apple (fruit) ()
  (:default-initargs :color 'red)
  (:default-initargs :price 2))
(make-instance 'apple :digestable 'no)
```

Συγκρίνοντας τα πλαίσια με τα σημασιολογικά δίκτυα μπορούμε να σημειώσουμε τα παρακάτω:

- Τα πλαίσια και τα σημασιολογικά δίκτυα αναπαριστούν παρόμοια γνώση.
- Ένα πλαίσιο μετατρέπεται αυτόματα σε ένα σημασιολογικό δίκτυο και αντίστροφα.
- Οι διαφορές αφορούν περισσότερο την τυπική χρήση:
 - Τα σημασιολογικά δίκτυα κανονικά θεωρούνται προδιαγραφές και δεν επιτρέπουν εξαιρέσεις ή προεπιλογές
 - Για τα πλαίσια που κανονικά θεωρούνται τυπικές περιγραφές, αναμένονται προεπιλογές (defaults) και εξαιρέσεων (excerptions).
- Τα σημασιολογικά δίκτυα συνήθως κάνουν έντονη διάκριση μεταξύ των κλάσεων και των στιγμιότυπων. Στα πλαίσια μια τέτοια διάκριση δεν είναι απαραίτητη δεδομένου ότι τα πλαίσια στιγμιότυπων γίνονται συνήθως συγκεκριμένα (παίρνουν πραγματικές τιμές σε επίπεδο slots) όταν μέσα σε αυτά αποτυπωθεί ο τρέχων κόσμος του προβλήματος και η περιγραφή τους δε διαφέρει σε τίποτα από την περιγραφή των πλαισίων ανώτερης κλάσης που είναι πάντα αφηρημένα.

Το ποιο είναι προτιμότερο για την αναπαράσταση της γνώσης ενός προβλήματος εξαρτάται από το είδος του προβλήματος και από τον επιστημονικό τομέα στον οποίο αυτό ανήκει.

Σενάρια

Τα **σενάρια** (scripts) είναι μία τεχνική που προτάθηκε από τον R. Schank (1975) στο πλαίσιο της σχετικής θεωρίας του περί πλαισίων (script theory) που εστιάζεται στη δόμηση της γνώσης στο πλαίσιο της κατανόησης της γλώσσας. Ο Schank υποστηρίζει ότι ένα σενάριο προορίζεται, κυρίως, για να εξηγήσει την επεξεργασία της γλώσσας και τις υψηλού επιπέδου δεξιότητες σκέψης. Τα σενάρια αποτελούν μία ειδική περίπτωση σχημάτων για την αναπαράσταση επεισοδιακής κυρίως γνώσης μέσω της παράθεσης πλαισίων σε χρονική σειρά.

Προκειμένου η θεωρία να αποδειχθεί, έχει αναπτυχθεί μια ποικιλία προγραμμάτων ηλεκτρονικών υπολογιστών. Ο Schank (1991) εφαρμόζει το θεωρητικό του πλαίσιο στην αφήγηση ιστοριών και στην ανάπτυξη ευφών εκπαιδευτικών προγραμμάτων. Ένα σενάριο περιγράφει μία στερεότυπη, αναμενόμενη ακολουθία γεγονότων με ειδικό περιεχόμενο και τις σχετικές με αυτά πληροφορίες χρησιμοποιώντας μία χρονική σειρά από πλαίσια και ειδικότερα:

- συνθήκες εισόδου στο σενάριο, δηλαδή συνθήκες που πρέπει να ικανοποιηθούν, πριν συμβούν τα γεγονότα του σεναρίου,
- αποτελέσματα που θα προκύψουν, αφού συμβούν τα γεγονότα του σεναρίου,
- ιδιότητες, δηλαδή θέσεις που αναπαριστούν αντικείμενα που περιέχονται στο σενάριο,
- ρόλους, δηλαδή θέσεις που αναπαριστούν οντότητες (π.χ. άνθρωποι) οι οποίες ενεργούν μέσα στο σενάριο,
- κανάλια που αποτελούν συγκεκριμένες περιπτώσεις γενικότερων προτύπων που περιγράφονται με ένα σενάριο. Διαφορετικά κανάλια του ίδιου σεναρίου θα μοιράζονται πολλά αλλά όχι όλα τα στοιχεία του.
- σκηνές που αποτελούν ακολουθίες γεγονότων που συμβαίνουν.

Παράδειγμα σεναρίου: Σενάριο Εστιατορίου (παρεμφερές: σενάριο καφετέριας)

Η είσοδος στο σενάριο απαιτεί ο πελάτης να πεινάει και να έχει χρήματα, για να πληρώσει το γεύμα του. Το σενάριο ολοκληρώνεται, όταν ο πελάτης δεν πεινάει πλέον, δηλαδή έχει φάει όσο επιθυμεί, και έχει πληρώσει τον ιδιοκτήτη του εστιατορίου. Ως αντικείμενα δηλώνονται εκείνα τα αντικείμενα του εστιατορίου που πρόκειται να χρησιμοποιήσει ο πελάτης, καθώς και τα χρήματα που διαθέτει:

Σενάριο: Εστιατόριο

Παραπομπή: Καφετέρια, Μαγειρέμα

Συνθήκες Εισόδου:

Πελάτης πεινάει

Πελάτης έχει χρήματα

Συνθήκες Εξόδου:

Πελάτης δεν πεινάει

Πελάτης έχει λιγότερα χρήματα

Ταμίας έχει περισσότερα χρήματα

Ιδιότητες / Αντικείμενα

τραπέζι

κατάλογος φαγητών (Μενού)

παραγγελία

φαγητό

λογαριασμός

φιλοδώρημα

χρήματα

Ρόλοι:

Πελάτης

Σερβιτόρος

Μάγειρας

Ταμίας

Κανάλια:

PTRANS: Μετακίνηση

ATTENDS: Στρέψη προσοχής

MBUILD: Αλλαγή στάσης

MTRANS: Κίνηση μέλους

ATRANS: Παράδοση αντικειμένου

INGEST: Κατανάλωση τροφής

COMPUTE: Υπολογισμός λογαριασμού

Σκηνή 1: Είσοδος

Πελάτης **PTRANS** Πελάτη στο εστιατόριο

Πελάτης **ATTENDS** μάτια στα τραπέζια

Πελάτης **MBUILD** που θα καθίσει
Πελάτης **PTRANS** στο τραπέζι

Σκηνή 2: Παραγγελία

(Μενού στο Τραπέζι)

Πελάτης **MTRANS** (Πελάτη παίρνει Μενού)

Πελάτης **MBUILD** επιλογή φαγητού

Πελάτης **MTRANS** σήμα στο Σερβιτόρο

Σερβιτόρος **PTRANS** Σερβιτόρο στο Τραπέζι

Πελάτης **MTRANS** παραγγελία στο Σερβιτόρο

Σερβιτόρος **PTRANS** Σερβιτόρος στην κουζίνα

Σερβιτόρος **ATRANS** παραγγελία στο Μάγειρα

Μάγειρας **Μαγειρέμα** παραγγελίας (Σενάριο προετοιμασίας φαγητού)

Σκηνή 3: Φαγητό

Μάγειρας **ATRANS** φαγητό στο Σερβιτόρο

Σερβιτόρος **ATRANS** φαγητό στον Πελάτη

Πελάτης **INGEST** φαγητό

(Επιλογή: Σκηνή 2 για επόμενη παραγγελία ή αλλιώς Σκηνή 4)

Σκηνή 4: Έξοδος

Σερβιτόρος **COMPUTE** (ετοιμάζει λογαριασμό)

Σερβιτόρος **PTRANS** λογαριασμό στο Τραπέζι

Σερβιτόρος **ATRANS** λογαριασμό στον Πελάτη

Πελάτης **ATRANS** φιλοδώρημα στο Σερβιτόρο

Πελάτης **PTRANS** Πελάτης στην έξοδο

Πελάτης **ATRANS** χρήματα στον Ταμία

Πελάτης **PTRANS** Πελάτης έξω από το εστιατόριο

Κανόνες

Οι **κανόνες** (rules) αποτελούν μία μέθοδο αναπαράστασης διαδικαστικής γνώσης που αντιστοιχεί σε έγκυρους συλλογισμούς και έχει τη μορφή **AN-TOTE**. Η δήλωση ή το σύνολο των δηλώσεων μετά τον όρο **AN** αντιπροσωπεύει κάποιο παρατηρήσιμο πρότυπο. Η δήλωση ή το σύνολο των δηλώσεων μετά τον όρο **TOTE** αντιπροσωπεύει κάποιο εξαγωγίμο συμπέρασμα ή κάποια εκτελεστέα ενέργεια.

Επομένως, ένα κανόνας:

- προσδιορίζει ένα πρότυπο και εξάγει συμπεράσματα σχετικά με το τι αυτό σημαίνει ή
- προσδιορίζει ένα πρότυπο και συμβουλεύει τι πρέπει να γίνει γι' αυτό ή
- προσδιορίζει ένα μοτίβο και πραγματοποιεί τις κατάλληλες ενέργειες.

Υπάρχουν αρκετοί διαφορετικοί όροι για τις δηλώσεις που έρχονται μετά το **AN** και όσα ακολουθούν μετά το **TOTE**. Οι δηλώσεις μετά το **AN** μπορεί να ονομάζονται συνθήκες, ισχυρισμοί, προϋποθέσεις, πρόγονοι, ενώ αυτά που ακολουθούν μετά το **TOTE** μπορούν να ονομάζονται συμπεράσματα, συνεπαγόμενα ή ενέργειες. Τα παραπάνω αποτυπώνονται στην τυπική μορφή ενός κανόνα που ακολουθεί:

Μορφή AN/IF-TOTE/THEN κανόνων:

IF A	<i>premises (ισχυρισμοί)</i>
and B	<i>antecedents (πρόγονοι, προηγούμενα)</i>
and ...	<i>preconditions (προϋποθέσεις)</i>
THEN X	<i>conclusions (συμπεράσματα)</i>
and Y	<i>consequents (συνεπαγόμενα)</i>

and ... actions (ενέργειες)

Οι κανόνες είναι τριών μορφών

- **συνεπαγωγικοί κανόνες** (deductive rules): το μέρος **IF** αυτών δηλώνει προηγούμενα των οποίων η αλήθεια οδηγεί σε συμπεράσματα για το γύρω κόσμο, χωρίς να τον αλλάζουν.
- **κανόνες παραγωγής** (productive rules): η αλήθεια των ισχυρισμών του **IF** μέρους οδηγεί σε ενέργειες που αλλάζουν το γύρω κόσμο μέσω της παραγωγής νέας γνώσης.
- **ενεργοί κανόνες** (active rules): για να ελεγχθεί η αλήθεια των προϋποθέσεων του μέρους **IF** και να εκτελεστούν οι ενέργειες του μέρους **THEN**, πρέπει πρώτα να συμβεί ένα συγκεκριμένο γεγονός.

Μορφές Κανόνων	Επεξήγηση
IF προηγούμενα THEN συνεπαγόμενα Συνεπαγωγικός κανόνας (Deductive rule)	AN τα προηγούμενα αληθεύουν TOTE αληθεύουν και τα συνεπαγόμενα
IF ισχυρισμοί THEN ενέργειες διαχείρισης γνώσης Κανόνας Παραγωγής (Production rule)	AN οι ισχυρισμοί αληθεύουν TOTE ενέργησε για να παραχθεί νέα γνώση
ON συμβάν IF προϋποθέσεις THEN ενέργειες Ενεργός κανόνας (Action rule)	Όταν συμβεί το γεγονός (συμβάν) AN οι προϋποθέσεις αληθεύουν TOTE εκτέλεσε τις ενέργειες

Πίνακας 2.3 Μορφές Κανόνων

Παράδειγμα συνεπαγωγικού κανόνα:

AN ο εκτυπωτής τυπώνει σωστά
και τα χρώματα δεν τυπώνονται σωστά
TOTE έχει τελειώσει το έγγραφο μελάνι

Παράδειγμα Κανόνα Παραγωγής:

AN ο εκτυπωτής τυπώνει σωστά
και τα χρώματα δεν τυπώνονται σωστά
TOTE αλλάξτε την κεφαλή με το έγγραφο μελάνι

Παράδειγμα Ενεργού κανόνα:

Όταν φέρει ο πελάτης έναν εκτυπωτή για έλεγχο
AN ο πελάτης είναι νέος
TOTE ενημέρωσέ τον για το κοστολόγιο του ελέγχου

Τα **συστήματα που βασίζονται σε κανόνες** ή **συστήματα κανόνων** (rule-based systems) είναι πολύ δημοφιλή, διότι:

- Είναι εύκολη, ακόμη και για μη ειδικούς, η ανάγνωση και κατανόηση των κανόνων, διότι είναι δομημένοι ως συλλογισμοί, δηλαδή με τρόπο παρόμοιο με αυτόν που οι άνθρωποι χρησιμοποιούν, για να δώσουν λογικές εξηγήσεις κατά την επίλυση προβλημάτων.
- Οι κανόνες είναι εύκολοι στη διαχείρισή τους για την προσθήκη νέων και μεταβολή ή διαγραφή παλαιών.
- Είναι εύκολο να δίνονται επεξηγήσεις για την πορεία της λύσης, απλώς με παρακολούθηση της σειράς με την οποία εκτελούνται οι κανόνες. .

Το κυριότερο μειονέκτημα των συστημάτων κανόνων είναι η απαίτηση ύπαρξης ενός πολύ αποτελεσματικού μηχανισμού αναζήτησης για την εύρεση του κατάλληλου κανόνα προς εκτέλεση σε κάθε κύκλο λειτουργίας τους. Άλλο μειονέκτημα είναι η μη ευέλικτη σύνταξη των κανόνων, που καθιστά δύσκολη την αναπαράσταση γεγονότων και αλγοριθμικών πληροφοριών.

Οντολογίες

Μια **οντολογία** (ontology) είναι μια τυπική αναπαράσταση ενός συνόλου εννοιών εντός ενός τομέα και των σχέσεων μεταξύ των εννοιών αυτών. Σε περιπτώσεις όπου η γνώση χρειάζεται να μοντελοποιηθεί, να δομηθεί και να διασυνδεθεί, οι οντολογίες παρέχουν έναν ικανοποιητικό μηχανισμό αναπαράστασης και διαμοίρασής της σε ομάδες ανθρώπων.

Η χρήση οντολογιών βοηθά στην ανάπτυξη του Παγκόσμιου Σημασιολογικού Ιστού (Semantic Web) ως το επόμενο στάδιο μετά τον Παγκόσμιο Ιστό (World Wide Web - WWW), δεδομένου ότι η ταξινόμηση που διαθέτει με τη βοήθεια απλών μηχανισμών συλλογιστικής καθιστά εφικτή την αναζήτηση με βάση έννοιες αντί λέξεις-κλειδιά που επιτρέπουν τη σημασιολογική εστίαση/διεύρυνση των ερωτήσεων, τις ερωταποκρίσεις ως προς περισσότερους του ενός όρους και τη χρήση τελεστών μετασχηματισμού των κειμένων. Τα παραπάνω επιτυγχάνονται, διότι οι οντολογίες ορίζουν δύο ουσιώδη συστατικά τα οποία συμβάλλουν στην πλήρη ανάπτυξη του Παγκόσμιου Ιστού:

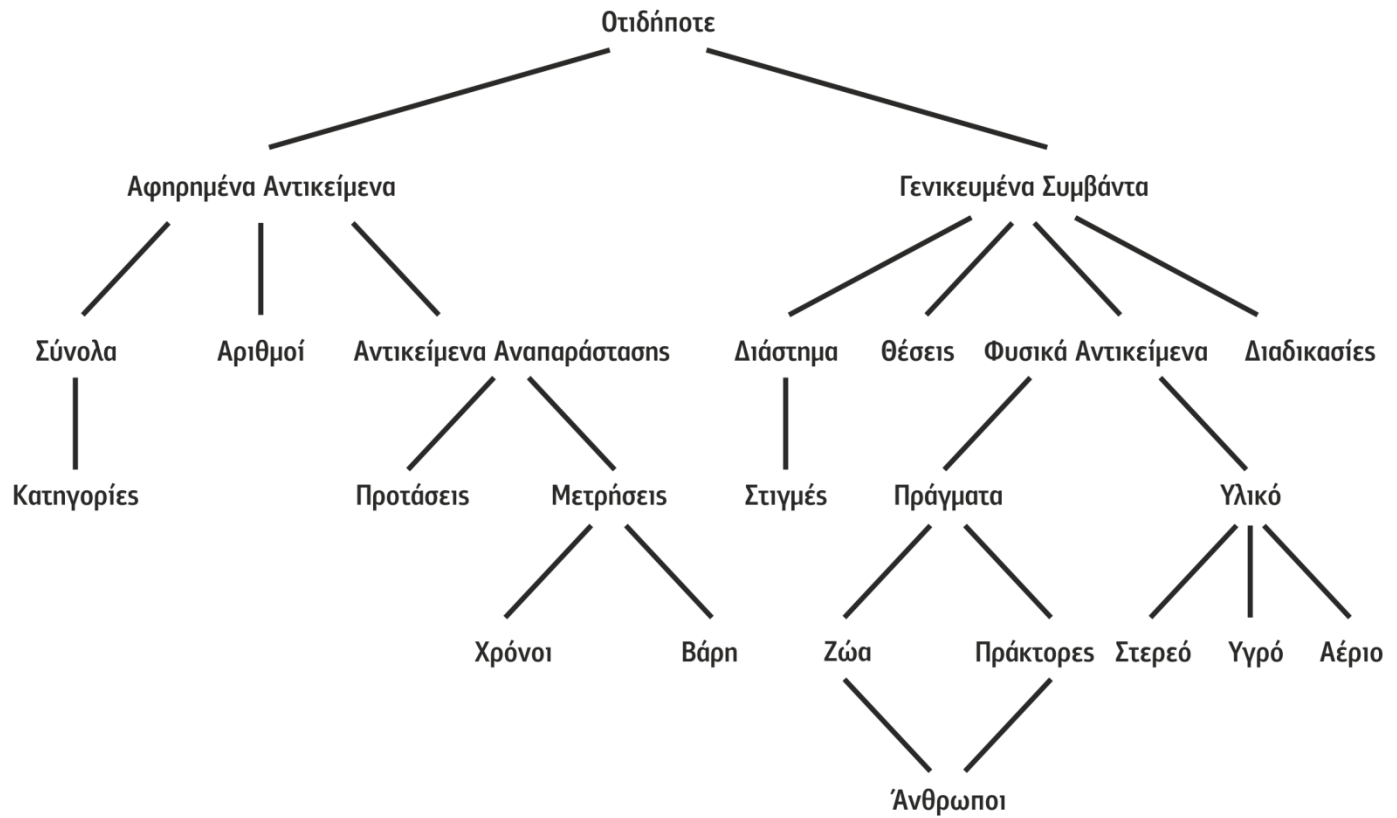
- την τυπική σημασιολογία της πληροφορίας, οπότε διευκολύνεται την επεξεργασία της πληροφορίας από τον Η/Υ,
- τη σημασιολογία του πραγματικού κόσμου, οπότε επιτρέπεται τη σύνδεση του περιεχομένου το οποίο υφίσταται επεξεργασία μηχανικά, με τη σημασία που του δίνουν οι άνθρωποι βασιζόμενοι σε κοινά αποδεκτή ορολογία.

Η έννοια της οντολογίας έχει υιοθετηθεί από την Τεχνητή Νοημοσύνη και σημαίνει:

«μια διαμοιρασμένη και κοινή κατανόηση κάποιου τομέα, η οποία μπορεί να ανταλλάξει μεταξύ ανθρώπων και συστημάτων εφαρμογών» (Gruber, 2009)

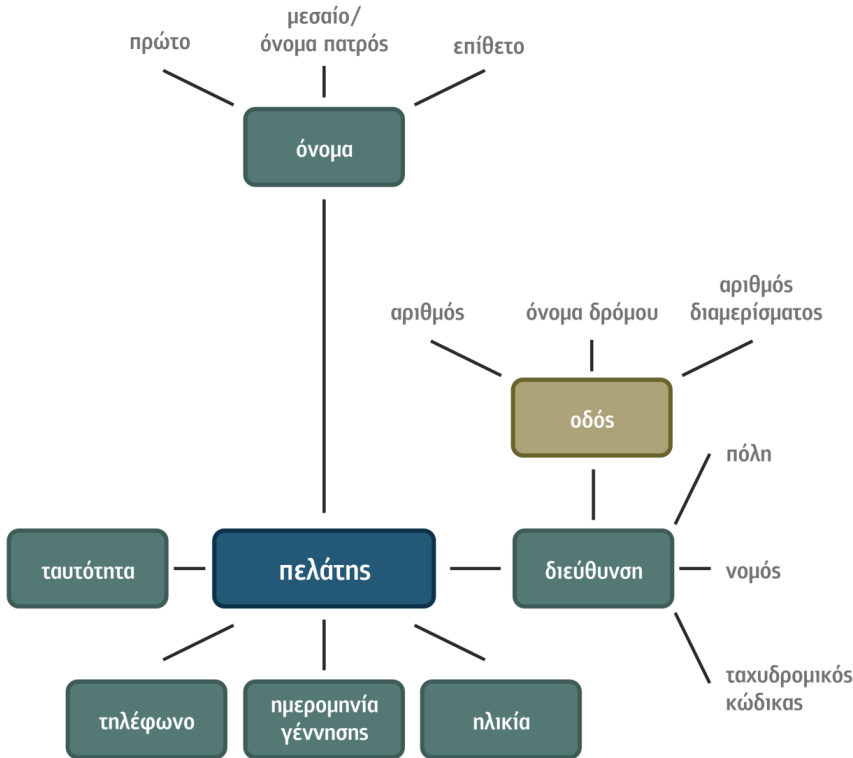
Για να αποτυπωθεί μια οντολογία, μπορούν να υιοθετηθούν και λογικές αναπαραστάσεις και δομημένες αναπαραστάσεις από αυτές που παρουσιάστηκαν στις προηγούμενες παραγράφους.

Σήμερα η τεχνολογία των οντολογιών χρησιμοποιείται από την ΤΝ για αναπαράσταση σφαιρικής γνώσης και αποτελεί γενική αναπαράσταση της γνώσης για τον κόσμο ως σύνολο μέσω αναπαράστασης όλων των γενικών εννοιών, π.χ. φυσικά-αφηρημένα-σύνθετα αντικείμενα, ιδιότητες, ιεραρχία, δομή, καταστάσεις, πεποιθήσεις, ενέργειες, προϋποθέσεις, επακόλουθα, χώρος, χρόνος, αλλαγές, διαδικασίες, συναρτήσεις, σχέσεις κτλ. (βλέπε Σχήμα 2.12)



Σχήμα 2.12 Γραφική αναπαράσταση σφαιρικής οντολογίας

Μπορεί, όμως, να χρησιμοποιηθεί και για αναπαράσταση ειδικής γνώσης, οπότε καλείται *οντολογία ειδικού σκοπού* που είναι εφαρμόσιμη σε κάθε δυνατό πεδίο του πραγματικού κόσμου με την προσθήκη κατάλληλων αξιωμάτων πεδίου (βλέπε Σχήμα 2.13).



Σχήμα 2.13 Παράδειγμα γραφικής αναπαράστασης οντολογίας ειδικού σκοπού

Μια οντολογία διαθέτει κατηγορίες (π.χ. η κατηγορία *cube*) που μπορούν να αναπαρασταθούν με προτασιακή λογική ως ιδιότητες αντικειμένων, π.χ. *cube (b1)*, ή ως συγκεκριμένο αντικείμενο του κόσμου του προβλήματος, π.χ. *is (b1 cube)*. Υποστηρίζει την *ταξινόμηση (taxonomy)* για την αντιμετώπιση του μεγάλου όγκου αντικειμένων ορίζοντας υποκατηγορίες, ιεραρχική δομή κατηγοριών και υποκατηγοριών και *κληρονομικότητα (inheritance)*, όπου τα στιγμιότυπα μιας υποκατηγορίας κληρονομούν τις ιδιότητες της γονικής κατηγορίας και πιθανόν να έχουν και επιπλέον ιδιότητες, δομώντας με τον τρόπο αυτόν ένα σημασιολογικό δίκτυο με σχέσεις ΑΚΟ και ΙΣΑ.

Μία οντολογία μπορεί να διατυπωθεί με κατηγορηματική λογική πρώτης τάξης, για να ορίσει κατηγορίες, αντικείμενα ως μέλη κατηγοριών, υποκατηγορίες μιας κατηγορίας, ιδιότητες κοινές στα μέλη μιας κατηγορίας και αναγνώριση μελών μιας κατηγορίας βάσει των κοινών χαρακτηριστικών τους.

Τα βασικά συστατικά μίας οντολογίας είναι πέντε:

- **Κλάσεις (classes):** έννοιες που σχετίζονται με ένα πεδίο ή κάποιες εργασίες, οι οποίες είναι συνήθως οργανωμένες σε κάποιο ταξινομικό σύστημα, παράδειγμα: σε μια οντολογία που αφορά το πανεπιστήμιο ο «φοιτητής» και ο «καθηγητής» αποτελούν δύο κλάσεις.
- **Σχέσεις (relations):** Ένας τύπος αλληλεπίδρασης μεταξύ εννοιών ενός πεδίου, όπως *a_kind-of* και *is-a*.
- **Συναρτήσεις (functions):** μια ειδική περίπτωση σχέσης στην οποία το *n*-οστό στοιχείο της σχέσης προσδιορίζεται μοναδικά από τα *n-1* προηγούμενα στοιχεία. παράδειγμα: Η τιμή-μεταχειρισμένου-αυτοκινήτου μπορεί να προσδιορίζεται ως συνάρτηση της αρχικής τιμής του καινούριου αυτοκινήτου, του μοντέλου του αυτοκινήτου, των χαρακτηριστικών του αυτοκινήτου και των χιλιομέτρων που έχει διανύσει.
- **Αξιώματα (axioms):** αναπαριστούν προτάσεις που είναι πάντα αληθείς. παράδειγμα: Όταν είναι κάποιος φοιτητής, οφείλει να παρακολουθεί τα μαθήματά του.

- **Στιγμιότυπα** (instances) : αναπαριστούν συγκεκριμένα στοιχεία.
παράδειγμα: ο φοιτητής με το όνομα Νίκος (είναι ένα στιγμιότυπο της κλάσης “φοιτητής”).

Βιβλιογραφία/Αναφορές

- Ausubel, D. P. (1967). *Learning Theory and Classroom Practice*. Οντάριο: The Ontario Institute For Studies In Education.
- Ausubel, D. P. (1968). *Educational Psychology. A Cognitive View*. Νέα Υόρκη: Holt, Rinehart & Winston, Inc.
- Cañas, A.J., & Novak, J. D. (2006). Re-examining the foundations for effective use of concept maps. Στο A. J. Cañas & J. D. Novak (επιμ.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the Second International Conference on Concept Mapping* (τόμος 1, σ. 494-502). Σαν Χοσέ, Κόστα Ρίκα: Universidad de Costa Rica. Ανακτήθηκε από http://www.ihmc.us/groups/acanas/wiki/bcb07/Alberto_J_Caas_Publications.html
- Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8 (2), 240–247. Ανακτήθηκε από <http://matt.colorado.edu/teaching/categories/cq69.pdf>
- Collins, A. M., & Quillian, M. R. (1970). Does category size affect categorization time?. *Journal of Verbal Learning and Verbal Behavior*, 9 (4), 432–438. Περισσότερες δημοσιεύσεις του Α.Μ. Collins στο <http://allancollins.northwestern.edu/pubs-more.html>
- Gruber, T. (2009). Ontology . Στο L. Liu & M. T. Özsu (επιμ.), *The Encyclopedia of Database Systems*. Νέα Υόρκη: Springer-Verlag. Ανακτήθηκε από <http://tomgruber.org/writing/ontology-definition-2007.htm>
- Minsky, M. (1975). A framework for representing knowledge. Στο P. H. Winston (επιμ.), *The Psychology of Computer Vision* (σ. 211-277). New York: McGraw-Hill. Ανακτήθηκε από <http://courses.media.mit.edu/2004spring/mas966/Minsky1974Frameworkforknowledge.pdf>.
- Novak, J.D., & Gowin, D.B. (1984). *Learning How to Learn*. Νέα Υόρκη & Καίμπριτζ, Η.Β: Cambridge University Press.
- Novak, J.D. (1991). Clarify with Concept Maps: A tool for students and teachers alike. *The Science Teacher*, 58 (7), 45-49.
- Rumelhart, D. E., & Ortony, A. (1977). The representation of knowledge in memory. Στο R. C. Anderson and R. J. Spiro (επιμ.), *Schooling and the Acquisition of Knowledge* (σ. 99-135) . Hillsdale, Νιου Τζέρσεϊ: Erlbaum. Ανακτήθηκε από http://www.cs.northwestern.edu/~ortony/Andrew_Ortony.html.
- Rumelhart, D. E. (1975). Notes on a schema for stories. Στο D. Bobrow & A. Collins (επιμ.), *Representation and understanding* (σ. 237-272). Νέα Υόρκη: Academic Press.
- Rumelhart, D. E. (1980). Schemata: The building blocks of cognition. Στο R. J. Spiro, B. C. Bruce, & W. F. Brewer (επιμ.), *Theoretical Issues in Reading Comprehension* (σ. 33-58). Hillsdale, Νιου Τζέρσεϊ: Erlbaum.
- Schank, R.C. (επιμ.) (1975). *Conceptual Information Processing*. Νέα Υόρκη : Elsevier.
- Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding: An inquiry into human knowledge structures*. Hillsdale, Νιου Τζέρσεϊ : Erlbaum Assoc.

Schank, R.C. (1991). *Tell Me a Story: A New Look at Real and Artificial Intelligence*. Νέα Υόρκη: Simon & Schuster.

Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3^η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστήμιου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Σε τι διαφέρει μια μέθοδος αναπαράστασης γνώσης από μια γλώσσα προγραμματισμού (π.χ. Java, C ++);

Α. Οι μέθοδοι αναπαράστασης μπορούν να χρησιμοποιηθούν για να εκφράσουν καταστάσεις (states), ενώ οι γλώσσες προγραμματισμού για να κωδικοποιήσουν συγκεκριμένες καταστάσεις με βάση την αναπαράσταση που έχει επιλεγεί.

Β. Οι γλώσσες προγραμματισμού μπορούν να χρησιμοποιηθούν για να δηλώσουν καταστάσεις (states) στις οποίες μπορεί να βρεθεί ένα πρόβλημα κατά την επίλυσή του ανεξάρτητα με τη μορφή αναπαράστασης που έχει επιλεγεί για την περιγραφή του.

Γ. Οι γλώσσες προγραμματισμού ταυτίζονται με αντίστοιχες μεθόδους αναπαράστασης.

Απάντηση/Λύση

Η σωστή απάντηση είναι η Α.

Κριτήριο αξιολόγησης 2

Αντιστοιχίστε τους όρους με τους ορισμούς :

Α. Δηλωτική Γνώση	Γνώση για τις δεξιότητες που απαιτούνται, για να εκτελέσουμε μια εργασία.
Β. Διαδικαστική Γνώση	Γνώση που έχουμε για τα γεγονότα και τα συμβάντα ενός κόσμου.

Απάντηση/Λύση

Το Α ταιριάζει με το 2 και το Β με το 1.

Κριτήριο αξιολόγησης 3

Ποια είναι τα χαρακτηριστικά των φυσικών γλωσσών (π.χ. ελληνικά, αγγλικά)

Σημειώστε τις σωστές απαντήσεις:

Α. Οι φυσικές γλώσσες έχουν εξελιχθεί, για να καλύψουν τις ανάγκες της επικοινωνίας.

Β. Οι φυσικές γλώσσες έχουν εξελιχθεί, για να καλύψουν τις ανάγκες της αναπαράστασης μιας έννοιας.

Γ. Το νόημα μιας πρότασης εξαρτάται από την ίδια την πρόταση και το πλαίσιο εντός του οποίου η πρόταση αυτή χρησιμοποιείται.

Δ. Ο διαμοιρασμός της γνώσης γίνεται χωρίς τη ρητή αναπαράσταση της ίδιας της γνώσης.

Ε. Δεν επιτρέπεται να είναι διφορούμενη, γιατί δημιουργείται πρόβλημα στην κατανόησή της.

Απάντηση/Λύση

Οι σωστές απαντήσεις είναι οι Α, Γ, Δ.

Κριτήριο αξιολόγησης 4

Τοποθετήστε τα επίπεδα της πυραμίδας της γνώσης σε σειρά από τη βάση προς την κορυφή:

- A. γνώση,
- B. σοφία,
- Γ. δεδομένα,
- Δ. πληροφορία.

Απάντηση/Λύση

Η σωστή σειρά είναι Γ, Δ, Α, Β

Κριτήριο αξιολόγησης 5

Ποια από τα παρακάτω αποτελούν γνωστά σχήματα αναπαράστασης γνώσης;

- A. κανόνες,
- B. δένδρα αναζήτησης,
- Γ. συλλογισμοί,
- Δ. λογική,
- E. σημασιολογικά δίκτυα.

Απάντηση/Λύση

Είναι τα Α, Δ, E

Κριτήριο αξιολόγησης 6

Ποιο από τα παρακάτω είναι το βασικό χαρακτηριστικό που πρέπει να διαθέτει ένα επιλεγμένο σύστημα αναπαράστασης γνώσης;

- A. δυνατότητα αναζήτησης με αλγόριθμους,
- B. καταλληλότητα κωδικοποίησης,
- Γ. δυνατότητες εξαγωγής συμπερασμάτων.

Απάντηση/Λύση

Η σωστή απάντηση είναι το Γ.

ΚΕΦΑΛΑΙΟ 3 - Συστήματα Κανόνων

Σύνοψη

Στο κεφάλαιο αυτό παρουσιάζεται ο χώρος των συστημάτων κανόνων με επικέντρωση στα συστήματα παραγωγής. Η χρήση κανόνων για την αναπαράσταση της διαδικαστικής και επεισοδιακής γνώσης ενός προβλήματος είναι μία από τις βασικές προσεγγίσεις για την επίλυση προβλημάτων βασισμένων στη γνώση στο πλαίσιο της Τεχνητής Νοημοσύνης. Αναλύονται εν συντομία τα διαφορετικά μοντέλα λογικής σκέψης στο χώρο της συλλογιστικής, παρουσιάζονται τα χαρακτηριστικά των συστημάτων κανόνων και περιγράφεται αναλυτικά ο τρόπος λειτουργίας των συστημάτων παραγωγής. Μεγάλη σημασία δίνεται στη διεξοδική παρουσίαση παραδειγμάτων λειτουργίας της προς τα εμπρός και προς τα πίσω αλυσιδωτής εκτέλεσης κανόνων.

Προαπαιτούμενη γνώση

Αναπαράσταση γνώσης

3.1 Συλλογιστική

Συλλογισμός είναι η ικανότητα του ανθρώπου να χρησιμοποιεί τις γνώσεις του για την εξαγωγή συμπερασμάτων σχετικά με τον κόσμο μέσα στον οποίο υπάρχει, ακολουθώντας διαφορετικά μοντέλα λογικής σκέψης. Η ικανότητα συλλογισμού αποτελεί ένα βασικό χαρακτηριστικό αυτού που αποκαλούμε “νοημοσύνη” στον άνθρωπο.

Η **συλλογιστική** (reasoning), ως επιστημονικός χώρος, περιλαμβάνει τους διαφορετικούς τρόπους συλλογισμού (μοντέλα λογικής σκέψης) με τους οποίους χρησιμοποιούμε την προϋπάρχουσα γνώση για τη δημιουργία νέας γνώσης.

Υπάρχουν τρία διαφορετικά μοντέλα λογικής:

- η επαγωγή,
- η απαγωγή,
- η παραγωγή ή συνεπαγωγή.

3.1.1 Επαγωγή

Κατά την **επαγωγή** (induction) καταλήγουμε σε νέα γνώση από προϋπάρχουσα γνώση που έχει προκύψει από την εμπειρία μας σε παρόμοιες καταστάσεις. Η επαγωγή ορίζεται ως η γενίκευση από περιπτώσεις που είναι ήδη γνωστές, για να συναχθούν συμπεράσματα για περιπτώσεις που δεν έχουν ακόμα βιωθεί.

Επαγωγή: Κοινές ιδιότητες μελών \Rightarrow συμπεράσματα για κλάση

Για παράδειγμα, ένας επί πτυχίω φοιτητής γνωρίζει πολύ καλά πώς θα διεξαχθούν οι γραπτές εξετάσεις στα μαθήματα στα οποία εξετάζεται, βάσει της εμπειρίας που έχει αποκτήσει από αντίστοιχες εξετάσεις κατά τη φοίτησή του.

Ανάλογα, θα μπορούσαμε να κάνουμε τον παρακάτω συλλογισμό:

Κάθε ημέρα έως σήμερα ο ήλιος ανατέλλει από την ανατολή.

\Rightarrow

Πάντα ο ήλιος θα ανατέλλει από την ανατολή.

Ο παραπάνω συλλογισμός δείχνει αληθοφανής, αλλά σε παρόμοιους συλλογισμούς εμπεριέχεται ο κίνδυνος λανθασμένου συμπεράσματος, γιατί η εμπειρία μας δεν είναι ποτέ πλήρης. Στο επόμενο παράδειγμα, ο αντίστοιχος συλλογισμός ότι:

Όσα κοράκια έχω δει μέχρι τώρα στη ζωή μου είναι μαύρα.

\Rightarrow

Όλα τα κοράκια του κόσμου είναι μαύρα.

δεν είναι βέβαιο ότι είναι ορθός, δεδομένου ότι είναι πολύ πιθανό να υπάρχει κάπου στον κόσμο ένα είδος κορακιών διαφορετικού χρώματος που δεν έτυχε ποτέ να δω.

Παρόλα αυτά, η δυνατότητα επαγωγικών συλλογισμών αποτελεί βασικό τρόπο εκδήλωσης της νοημοσύνης του ανθρώπου που τον καθιστά ικανό να μαθαίνει από τις εμπειρίες του, ακόμα και αν σε ορισμένες περιπτώσεις καταλήξει σε λανθασμένα συμπεράσματα.

Στο χώρο της Τεχνητής Νοημοσύνης, η επαγωγή χρησιμοποιείται επιτυχώς από συστήματα μηχανικής μάθησης, στα οποία θα αναφερθούμε στο επόμενο [κεφάλαιο 4](#). Στο χώρο της μηχανικής μάθησης, ένας μηχανισμός εξαγωγής συμπερασμάτων που χρησιμοποιεί επαγωγικό συλλογισμό βασίζεται πάντα στη συλλογή παραδειγματικών περιπτώσεων και η εξαγωγή συμπερασμάτων προκύπτει από συγκέντρωση παρατηρήσεων που αφορούν κοινά χαρακτηριστικά των παραδειγματικών περιπτώσεων. Τα συμπεράσματα αυτά μπορεί να εξαχθούν ως νέα διαδικαστική γνώση υπό μορφή κανόνων.

Για παράδειγμα:

Παρατηρήσεις: Έχω καλές βαθμολογίες στα μαθήματα των οποίων παρακολούθησα τις παραδόσεις.

Κανόνας: Όταν παρακολουθώ τις παραδόσεις ενός μαθήματος, το περνώ με καλό βαθμό.

3.1.2 Απαγωγή

Η **απαγωγή** (abduction) συνάγει λογικά συμπεράσματα βασιζόμενη σε σχετικές μέχρι τώρα μαρτυρίες και χρησιμοποιείται όταν δεν διαθέτουμε ολοκληρωμένες πληροφορίες για το πρόβλημα που επιχειρούμε να επιλύσουμε. Στην ουσία είναι ένας τρόπος να κάνουμε την «καλύτερη λογική υπόθεση» με βάση σχετικές διαθέσιμες μαρτυρίες, όταν οι γνώσεις που διαθέτουμε είναι ελλιπείς.

Απαγωγή: *Μη πλήρης γνώση \Rightarrow εικασίες*

Για παράδειγμα:

Οι άντρες πάντα θυμώνουν, όταν κάποιος τους καθυστερεί. (διαθέσιμες μαρτυρίες)

Ο φίλος μου είναι θυμωμένος. (μη πλήρης γνώση, λείπει η αιτία)

Πιθανώς το ότι ο φίλος μου θύμωσε να οφείλεται στο ότι τον καθυστέρησα. (καλύτερη λογική υπόθεση-εικασία)

Ένας μηχανισμός εξαγωγής συμπερασμάτων που χρησιμοποιεί απαγωγικούς συλλογισμούς βασίζεται αφενός σε υπάρχοντα δεδομένα και σε γεγονότα που συνθέτουν το πρόβλημα και αφετέρου σε σχετικές διαθέσιμες μαρτυρίες. Για παράδειγμα, ο παραπάνω συλλογισμός μπορεί να παρουσιαστεί δομημένα και ως ακολούθως:

Διαθέσιμες μαρτυρίες: Οι άντρες πάντα θυμώνουν, όταν κάποιος αργεί να πάει σε μια συνάντηση.

Γεγονός: Ο φίλος μου είναι θυμωμένος.

Δεδομένο: Άργησα να πάω στη συνάντησή μας.

Συμπέρασμα: Ο φίλος μου είναι θυμωμένος, επειδή άργησα να πάω στη συνάντησή μας.

Γνωστά συστήματα ΤΝ που εξάγουν λογικά συμπεράσματα για ένα πρόβλημα χρησιμοποιώντας την προϋπάρχουσα εμπειρία από παρόμοιες περιπτώσεις είναι τα λεγόμενα συστήματα Συλλογιστικής Βασισμένης σε Περιπτώσεις (Case-Based Reasoning -ΣΒΠ/CBR) που αναλύονται στο κεφάλαιο της μηχανικής μάθησης **(να μπει kj στη συγκεκριμένη παράγραφο του κεφαλαίου 4)**.

3.1.3 Παραγωγή

Το τρίτο είδος συλλογισμού, η **παραγωγή** (deduction), βασίζεται στην τυπική λογική, όπως αυτή έχει παρουσιαστεί στο [προηγούμενο κεφάλαιο](#) περί αναπαράστασης γνώσης.

Παραγωγή: *Γνώση για την κλάση* \Rightarrow *συμπεράσματα για το μέλος*

Για παράδειγμα:

Αν βρέχει και κάθομαι κάτω από τη βροχή, θα βραχούν τα μαλλιά μου. (κανόνας)

Βρέχει και κάθομαι κάτω από τη βροχή. (γεγονότα)

\Rightarrow

Θα βραχούν τα μαλλιά μου. (λογικό συμπέρασμα που προκύπτει από τον κανόνα)

Η παραγωγή αποτελεί τον μόνο έγκυρο συλλογισμό, αλλά η εφαρμογή της δεν είναι απαραίτητο να οδηγήσει σε σωστά συμπεράσματα, αν ο συλλογισμός δεν είναι σωστά σχεδιασμένος. Λόγου χάρη, στο προηγούμενο παράδειγμα δεν είναι πάντα αληθές το συμπέρασμα, γιατί ο συλλογισμός έχει «κενά». Δηλαδή, δεν έχει προβλεφθεί από το σχεδιαστή του να περιληφθεί στους ισχυρισμούς ότι απαιτείται, επίσης, να μην κρατώ ομπρέλα, γιατί, αν κρατώ ομπρέλα, τα μαλλιά μου δε θα βραχούν, οπότε το συμπέρασμα καθίστανται λανθασμένο και ο κανόνας προβληματικός. Για το λόγο αυτόν, παρότι η παραγωγή είναι πάντα έγκυρος μηχανισμός εξαγωγής συμπεράσματος, πρέπει παράλληλα να γίνει σαφές ότι είναι μια τεχνική εξαγωγής συμπερασμάτων που δεν οδηγεί απαραίτητα σε σωστά συμπεράσματα, αν η γνώση που αποτυπώνει είναι ελλιπής ή λανθασμένη.

Γνωστά στο χώρο της συλλογιστικής είναι τα δύο μοντέλα λογικής σκέψης και έγκυροι παραγωγικοί συλλογισμοί τα **Modus Ponens** και **Modus Tollens**. Στο μεν πρώτο μοντέλο η αλήθεια των ισχυρισμών οδηγεί στην παραδοχή των συμπερασμάτων – έχουμε ήδη παρουσιάσει αρκετά σχετικά παραδείγματα παραπάνω. Στο δεύτερο μοντέλο, η άρνηση των συμπερασμάτων οδηγεί στην άρνηση των ισχυρισμών.

Για παράδειγμα:

Αν βρέχει και κάθομαι κάτω από τη βροχή, θα βραχούν τα μαλλιά μου. (κανόνας)

ΔΕΝ έχουν βραχεί τα μαλλιά μου. (άρνηση συμπεράσματος)

\Rightarrow

Η ΔΕΝ βρέχει ή ΔΕΝ κάθομαι κάτω από τη βροχή ή και τα δύο. (άρνηση ισχυρισμών ως λογικό συμπέρασμα)

Η εξαγωγή συμπερασμάτων σε κάθε ένα από τα δύο μοντέλα υλοποιείται ως ακολούθως:

Modus Ponens: *ΑΝ Α αληθές ΤΟΤΕ Β αληθές*

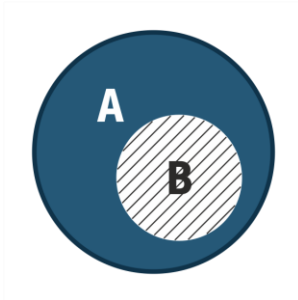
Α αληθές \Rightarrow Β αληθές



Σχήμα 3.1 Κύκλοι του Euler για το Modus Ponens

Δείτε [κινούμενη εικόνα 3.1 - Κύκλοι του Euler \(Modus Ponens\)](#)

Modus Tollens: *AN A αληθές, ΤΟΤΕ B αληθές*
B όχι αληθές \Rightarrow A όχι αληθές



Σχήμα 3.2 Κύκλοι του Euler για το Modus Tollens

Δείτε [κινούμενη εικόνα 3.2 - Κύκλοι του Euler \(Modus Tollens\)](#)

Βάσει όλων των παραπάνω, η υλοποίηση συγκεκριμένων συλλογιστικών μέσα στα συστήματα κανόνων προϋποθέτει την υιοθέτηση αντίστοιχων *μηχανισμών εξαγωγής συμπερασμάτων (inference mechanisms)* που βασίζονται σε:

- αναζήτηση οδηγούμενη από τους **στόχους** (goal-driven ή top-down),
- αναζήτηση οδηγούμενη από τα **δεδομένα** (data-driven ή bottom-up).

Εκτενέστερη αναφορά στους μηχανισμούς αυτούς θα γίνει στην παράγραφο περί συστημάτων παραγωγής (ανατρέξτε στην [παράγραφο 3.4](#) παρακάτω).

3.2 Συστήματα Κανόνων

Τα συστήματα κανόνων ανήκουν στο γενικότερο χώρο των **Συστημάτων Βασισμένων στη Γνώση - ΣΒΓ** (Knowledge-Based Systems - KBS), όπου είναι απαραίτητη η επιλογή κατάλληλης αναπαράστασης της διαδικαστικής γνώσης εκφρασμένης υπό μορφή έγκυρων συλλογισμών. Τα ΣΒΓ/KBS που χρησιμοποιούν κανόνες για την αναπαράσταση της διαδικαστικής γνώσης του προς επίλυση προβλήματος καλούνται **Συστήματα Βασισμένα σε Κανόνες** (Rule-Based Systems-ΣΒΚ/RBS) ή απλώς **Συστήματα Κανόνων**.

3.2.1 Υλοποίηση κανόνων

Η δομή ενός κανόνα αποτελείται από δύο τμήματα, το τμήμα των ισχυρισμών (τμήμα AN/IF) και το τμήμα των ενεργειών ή συμπερασμάτων (τμήμα ΤΟΤΕ/THEN)::

AN <ισχυρισμοί>
ΤΟΤΕ <ενέργειες>/<συμπεράσματα>

Γενικά, υπάρχουν δύο μεγάλες κατηγορίες εξειδικεύσεων αυτής της γενικής μορφής. Στην πρώτη κατηγορία, οι σχέσεις μεταξύ ισχυρισμών και γεγονότων δηλώνονται ρητά, οπότε μια βάση κανόνων μπορεί να παρασταθεί ως δίκτυο που παριστάνει τις αλληλεξαρτήσεις μεταξύ κανόνων και γεγονότων, όπως αυτό

του [σχήματος 3.6](#). Στην κωδικοποίηση των κανόνων της πρώτης αυτής κατηγορίας στις συνθήκες τα γεγονότα ορίζονται ρητά.

Ένα παράδειγμα κανόνα που ακολουθεί την παραπάνω σύνταξη είναι το εξής:

```
AN ισχύει <Ελένη πληροφορικός>
ΤΟΤΕ πρόσθεσε το συμπέρασμα <είναι Ελένη νοήμων>
```

Η δεύτερη κατηγορία στηρίζεται στην έννοια του **ταιριάσματος προτύπων** (pattern matching), του οποίου ένας ορισμός είναι ο ακόλουθος:

Ένα **πρότυπο** (pattern) **ταιριάζει** (matches) με ένα **γεγονός** (fact), αν υπάρχουν **προσδέσεις** (bindings) για τις μεταβλητές στο πρότυπο τέτοιες, ώστε, αν αντικαταστήσουμε τις μεταβλητές με τις προσδεδωμένες τιμές (ή προσδέσεις) τους, το γεγονός και το πρότυπο γίνονται συντακτικά ταυτόσημα.

Οι σχέσεις μεταξύ ισχυρισμών και γεγονότων σ' αυτές τις εξειδικεύσεις δε δηλώνονται ρητά, αλλά σχηματοποιούνται κατά τη διαδικασία εξαγωγής συμπερασμάτων. Μπορούμε να έχουμε χρήση μεταβλητών στους ισχυρισμούς αλλά και στα συμπεράσματα.

Ένα παράδειγμα κανόνα που ακολουθεί την παραπάνω σύνταξη είναι το εξής:

```
AN ισχύει <είναι ?X πληροφορικός>
ΤΟΤΕ εμφάνισε "Βρέθηκε ένας πληροφορικός"
```

Παρατηρήστε ότι η μεταβλητή στο συγκεκριμένο παράδειγμα, για να διασαφηνιστεί ως τέτοια, έχει όνομα που ο πρώτος χαρακτήρας είναι συμβατικά ο χαρακτήρας "?". Μια τέτοια προσέγγιση ακολουθεί το προγραμματιστικό περιβάλλον της Lisp. Σε άλλα περιβάλλοντα, όπως αυτό της Prolog ακολουθούνται άλλες συμβατικές προσεγγίσεις.

Επίσης, μεταβλητές μέσα σε πρότυπα μπορούν να παρουσιάζονται και στο then μέρος ενός κανόνα, αλλά μόνο αν αυτές έχουν ήδη ταυτοποιηθεί στο if μέρος του ίδιου κανόνα. Για τη μεταβλητή στο μέρος then δεν ακολουθείται εκ νέου η διαδικασία ταυτοποίησης, διότι η διαδικασία που έχει ήδη γίνει για την ίδια μεταβλητή στο if μέρος μεταδίδεται και στην αντίστοιχη μεταβλητή του then μέρους. Η διεργασία αυτή καλείται **ενοποίηση** (unification). Ακολουθεί σχετικό παράδειγμα:

Δήλωση: <είναι Κώστας πληροφορικός>

Κανόνας:

```
AN ισχύει <είναι ?X πληροφορικός>
ΤΟΤΕ πρόσθεσε το συμπέρασμα <είναι ?X νοήμων>
```

Στο παραπάνω παράδειγμα, όταν εκτελεστεί ο κανόνας, η μεταβλητή ?X στο if μέρος θα ταυτιστεί με τη λέξη Κώστας, ταύτιση που μεταδίδεται στην ?X του then μέρους μέσω της ενοποίησης και έτσι παράγεται το συμπέρασμα <είναι Κώστας νοήμων>. Ενοποιήσεις μπορούν να γίνουν και στο if μέρος ενός κανόνα όταν η ίδια μεταβλητή αναφέρεται μέσα σε δυο ή περισσότερες διαδοχικές υποθέσεις. Για παράδειγμα, στον επόμενο κανόνα, η ταύτιση του πρώτου ?X με το Κώστας θα μεταδοθεί και στο δεύτερο ?X και για να είναι αληθείς οι υποθέσεις του κανόνα θα πρέπει να υπάρχει εκτός της δήλωσης <είναι Κώστας πληροφορικός> και η δήλωση <έχει Κώστας δουλειά>.

Κανόνας:

```
AN ισχύει <είναι ?X πληροφορικός>
    Και <έχει ?X δουλειά>
ΤΟΤΕ πρόσθεσε το συμπέρασμα <είναι ?X νοήμων>
```

Όταν οι ισχυρισμοί είναι παραπάνω του ενός, τότε συνδέονται με τα λογικά συνδετικά **AND** και **OR**. Επίσης, οι σχέσεις μεταξύ ισχυρισμών και γεγονότων μπορεί να δηλώνονται ρητά για κάποιους από αυτούς και για άλλους να χρησιμοποιούνται μεταβλητές. Για παράδειγμα, ένας κανόνας που υλοποιεί στο πρόβλημα του αγρότη τον τελεστή μετάβασης του αγρότη με τη χίνα από τη μία όχθη στην απέναντι, χρησιμοποιώντας μεταβλητή, για να δηλωθεί άρρητα η έννοια της όχθης αφετηρίας και της απέναντι όχθης, θα μπορούσε να συνταχθεί ως ακολούθως:

```
AN <is Farmer ?side>
   και <is Goose ?side>
   και <opposite ?side ?otherside>
ΤΟΤΕ αφαίρεσε τα γεγονότα <is Farmer ?side>, <is Goose ?side>
   πρόσθεσε τα γεγονότα <is Farmer ?otherside>
                           <is Goose ?otherside>
```

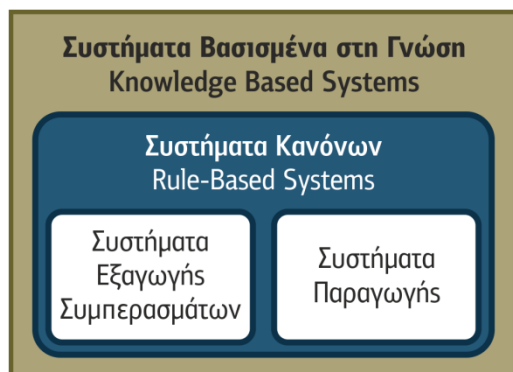
Στο παράδειγμα αυτό, αν υπάρχουν τα γεγονότα <is Farmer left> και <opposite left right> στη μνήμη εργασίας, η μεταβλητή ?side θα ταυτιστεί με το left και η μεταβλητή ?otherside θα ταυτιστεί με το right, σε όποιο σημείο του κανόνα και αν παρουσιάζονται αυτά. Συνεπώς, ο κανόνας μετά το ταίριασμα προτύπων και την ενοποίηση θα έχει μια μορφή που αντιστοιχεί σε αυτήν ενός κανόνα γραμμένου με ρητές αναφορές σε γεγονότα, όπως ο επόμενος:

```
AN <is Farmer left>
   και <is Goose left>
   και <opposite left right>
ΤΟΤΕ αφαίρεσε τα γεγονότα <is Farmer left>, <is Goose left>
   πρόσθεσε τα γεγονότα <is Farmer right>, <is Goose right>
```

3.2.2 Περιγραφή Συστημάτων Κανόνων

Τα συστήματα κανόνων διακρίνονται σε 2 κατηγορίες, ανάλογα με το αν ο σκοπός για τον οποίο χρησιμοποιούν την υπάρχουσα γνώση για ένα πρόβλημα είναι:

- να εξάγουν νέα συμπεράσματα που αφορούν τον κόσμο του προβλήματος, οπότε καλούνται **Συστήματα Εξαγωγής Συμπερασμάτων** (Deduction Systems).
- να διαφοροποιήσουν τον κόσμο του προβλήματος παράγοντας νέα σχετική γνώση δηλωτικής μορφής, οπότε καλούνται **Συστήματα Παραγωγής** (Production Systems).



Σχήμα 3.3 Συστήματα Βασισμένα στη Γνώση

Τα συστήματα κανόνων αποτελούνται από τρία κύρια μέρη :

- συλλογή γεγονότων,
- συλλογή κανόνων του τύπου AN-TOTE (IF-THEN),
- μηχανισμός εξαγωγής συμπερασμάτων (inference mechanism).

Η συλλογή γεγονότων περιγράφει τη δηλωτική γνώση για ένα πρόβλημα. **Γεγονός** (fact) είναι μία δήλωση ότι κάποιος ισχυρισμός είναι αληθής, όπως π.χ. «πέφτει χιόνι», «ο ασθενής έχει υψηλό πυρετό», «η γάτα μου είναι καλή σύντροφος». Η δηλωτική γνώση για το πρόβλημα προκύπτει από παρατηρήσεις που αφορούν τα αντικείμενα του κόσμου του προβλήματος και αποτυπώνεται κωδικοποιημένη σε μια συλλογή γεγονότων (facts) που καλείται **ενεργός μνήμη** (active memory) ή **μνήμη εργασίας** (working memory). Η μνήμη εργασίας περιέχει γεγονότα, τόσο αυτά που ορίζονται κατά την εκκίνηση του συστήματος όσο και αυτά που δημιουργούνται κατά την εκτέλεσή του.

Οι κανόνες αποτυπώνουν γνώση που προκύπτει από προϋπάρχουσα εμπειρία σε σχέση με το πρόβλημα. Σε έναν κανόνα στον οποίο υπάρχουν περισσότεροι του ενός ισχυρισμού στο IF μέρος, συνδεδεμένοι με AND και OR, η αλήθεια ή όχι αυτού του σύνθετου ισχυρισμού προκύπτει από τους νόμους της λογικής, όπως αυτοί αναπαριστώνται σε προηγούμενη παράγραφο (βλέπε [Σχήματα 3.1](#) και [Σχήμα 3.2](#)) ή [πίνακες αληθείας](#) ή άλλες μεθόδους.

Όταν το AN μέρος ενός κανόνα περιέχει έναν ή περισσότερους ισχυρισμούς, αυτοί πρέπει να είναι αληθείς, δηλαδή να περιέχονται στη συλλογή των γεγονότων, ώστε να μπορεί ο κανόνας να λειτουργήσει ως παραγωγικός συλλογισμός με αληθείς ισχυρισμούς και να δώσει έγκυρα αποτελέσματα ακολουθώντας το μοντέλο [modus ponens](#). Το TOTE μέρος περιέχει αποτελέσματα/συμπεράσματα που επιδρούν ή όχι στον κόσμο του προβλήματος, επιφέροντας ή όχι αλλαγές στην υπάρχουσα δηλωτική γνώση. Η συλλογή των κωδικοποιημένων κανόνων καλείται **Βάση Κανόνων** (Rules Base) και αποτελεί τη **Βάση Γνώσης** (Knowledge Base) του συστήματος.

Ο **μηχανισμός εξαγωγής συμπερασμάτων** (inference mechanism) ή **συμπερασματική μηχανή** (inference engine) είναι υπεύθυνος, αφού ελέγξει την αλήθεια των ισχυρισμών ενός κανόνα, να τον θέτει ως υποψήφιο προς εκτέλεση και να εξάγει τα συμπεράσματα μέσα στη μνήμη εργασίας, αν ο κανόνας επιλεγεί προς εκτέλεση. Η εκτέλεση ενός κανόνα καλείται **πυροδότηση κανόνα** (rule firing).

Η συμπερασματική μηχανή ελέγχει τη λειτουργία ολόκληρου του συστήματος και ενεργεί σε κύκλους λειτουργίας, όπου σε κάθε κύκλο πυροδοτείται μόνο ένας κανόνας.



Σχήμα 3.4 Τυπική Δομή και λειτουργία Συστημάτων Κανόνων

Κινούμενη εικόνα για το σχήμα 3.4

3.3 Συστήματα Εξαγωγής Συμπερασμάτων

Τα **συστήματα εξαγωγής συμπερασμάτων** (deduction systems) περιέχουν μη παραγωγική διαδικαστική γνώση, δηλαδή κανόνες που οδηγούν σε λογικά συμπεράσματα, αλλά δε διαφοροποιούν τον κόσμο του προβλήματος.

Οι κανόνες στα συστήματα εξαγωγής συμπερασμάτων καλούνται **συμπερασματικοί κανόνες** (inference rules) και εκφράζουν κυρίως δηλωτική γνώση:

ΑΝ οι ισχυρισμοί αληθεύουν
ΤΟΤΕ αληθεύει και το συμπέρασμα

Χαρακτηριστικό παράδειγμα είναι αυτό του γενεαλογικού δένδρου, όπου συνάγονται συμπεράσματα για τις συγγενικές σχέσεις μεταξύ των προσώπων που το απαρτίζουν, χωρίς αυτό να σημαίνει ότι υπάρχει κανόνας με την εκτέλεση του οποίου θα αλλάξει κάποια από τις σχέσεις μεταξύ των προσώπων αυτών.

Για παράδειγμα, έστω ότι σε ένα σύστημα εξαγωγής συμπερασμάτων έχουμε:

Μνήμη Εργασίας: {<X γονέας του Y>, <Y γονέας του Z>}
 Βάση Κανόνων: {R1:ΕΑΝ <X γονέας του Y> και <Y γονέας του Z>
 ΤΟΤΕ <X παππούς του Z>}

Όταν το παραπάνω σύστημα εκτελεστεί, η συμπερασματική μηχανή θα καλέσει τη *δομή ελέγχου* να εκτελέσει τον κανόνα R1, επειδή έχει αληθείς ισχυρισμούς. Η εκτέλεση θα προσθέσει το γεγονός <X παππούς του Z> στη μνήμη εργασίας, αλλά αυτό δε διαφοροποιεί τις υπάρχουσες συγγενικές σχέσεις μεταξύ των X, Y και Z· απλώς εξάγει κάποια συμπεράσματα που δεν είχαν συμπεριληφθεί ως γεγονότα στην αρχική μνήμη εργασίας.

Άλλα παραδείγματα συμπερασματικών κανόνων:

ΑΝ ένα ζώο διαθέτει τρίχωμα και το ζώο δίνει γάλα
 ΤΟΤΕ το ζώο είναι θηλαστικό

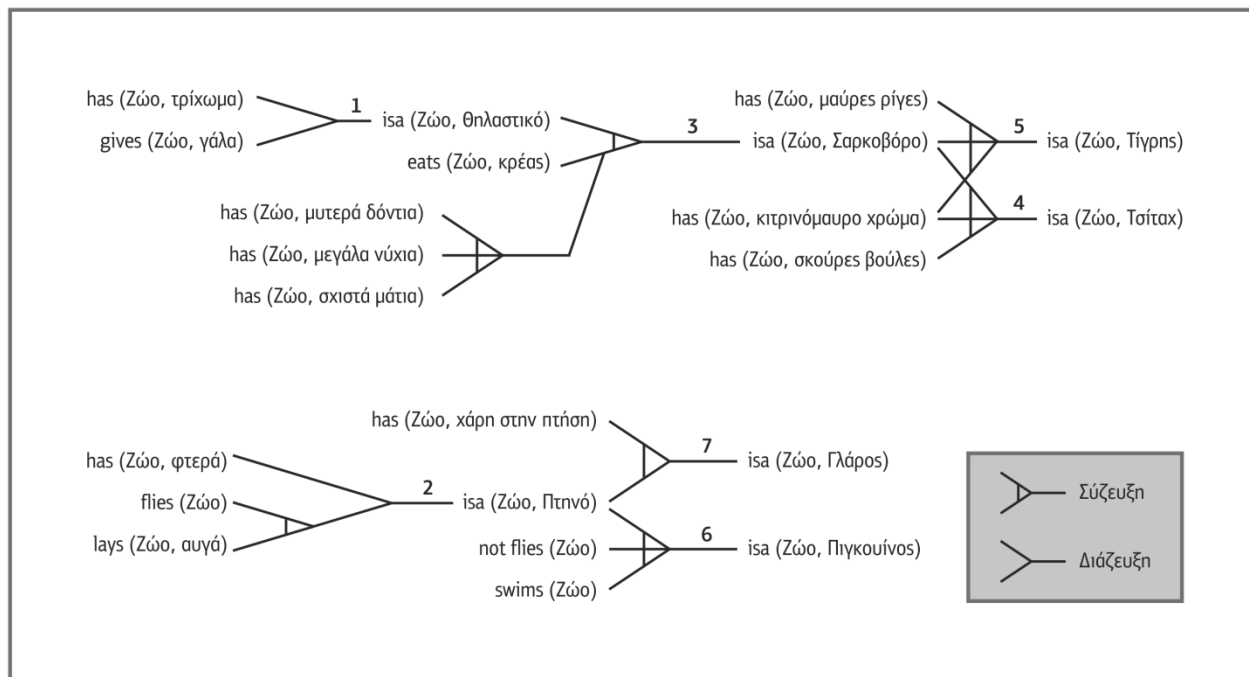
ΑΝ τα φώτα του δρόμου είναι αναμμένα
 ΤΟΤΕ υπάρχει ασφάλεια

Οι παρακάτω συμπερασματικοί κανόνες αποτυπώνουν δηλωτική γνώση από το ζωικό βασίλειο σε σχέση με τις ιδιότητες που χαρακτηρίζουν κάποια θηλαστικά και πτηνά. Η δομή τους περιέχει ισχυρισμούς και συμπεράσματα σε κατηγορηματική μορφή.

1: if has (Ζώο, τρίχωμα) or gives (Ζώο, γάλα) then isa (Ζώο, Θηλαστικό)	4: if has (Ζώο, κτρινόμαυρο χρώμα) and has (Ζώο, σκούρες βούλες) then isa (Ζώο, Τσίταχ)
2: if has (Ζώο, πούπουλα) or flies (Ζώο) and lays (Ζώο, αυγά) then isa (Ζώο, Πτηνό)	5: if isa (Ζώο, Σαρκοβόρο) and has (Ζώο, μαύρες ρίγες) then isa (Ζώο, Τίγρης)
3: if isa (Ζώο, Θηλαστικό) and eats (Ζώο, κρέας) or has (Ζώο, μυτερά δόντια) and has (Ζώο, μεγάλα δόντια) and has (Ζώο, σχιστά μάτια) then isa (Ζώο, Σαρκοβόρο)	6: if isa (Ζώο, Πτηνό) and not flies (Ζώο) and swims (Ζώο) then isa (Ζώο, Πγκουίνος)
	7: if isa (Ζώο, Πτηνό) and has (Ζώο, χάρη στην πτήση) then isa (Ζώο, Γλάρος)

Σχήμα 3.5 Σύστημα κανόνων μέρους ζωικού βασιλείου

Για τους παραπάνω κανόνες ισχύουν οι εξής συσχετισμοί που μπορούν να οδηγήσουν σε λογικά συμπεράσματα:



Σχήμα 3.6 Συσχετισμός κανόνων μέσα στη βάση γνώσης

Βάσει των συσχετισμών του σχήματος 3.6, αν υπάρχουν στη μνήμη εργασίας τα γεγονότα που ακολουθούν (καθιστώντας τους αντίστοιχους ισχυρισμούς των κανόνων αληθείς):

flies (Ιωνάθαν) ,
lays (Ιωνάθαν, αυγά) ,
has (Ιωνάθαν, χάρη στην πτήση)

τότε θα εξαχθούν λογικά τα παρακάτω συμπεράσματα:

isa (Ιωνάθαν, Πτηνό) και
isa (Ιωνάθαν, Γλάρος)

ακολουθώντας την εξής σειρά εξαγωγής λογικών συμπερασμάτων:

flies (Ιωνάθαν) ∨ lays (Ιωνάθαν, αυγά) → isa (Ιωνάθαν, Πτηνό)
isa (Ιωνάθαν, Πτηνό) ∨ has (Ιωνάθαν, χάρη στην πτήση)
→ isa (Ιωνάθαν, Γλάρος)

Κατά τα λοιπά, τα συστήματα εξαγωγής συμπερασμάτων λειτουργούν ακριβώς όπως και τα συστήματα παραγωγής που παρουσιάζονται στη συνέχεια.

3.4 Συστήματα παραγωγής

Στα **συστήματα παραγωγής** (production systems), η διαδικαστική γνώση για το πρόβλημα που επιχειρείται να επιλυθεί μέσω του συστήματος εκφράζεται με κανόνες της μορφής:

ΑΝ οι ισχυρισμοί αληθεύουν

ΤΟΤΕ εκτέλεσε τις ενέργειες που αλλάζουν τον κόσμο

Δηλαδή, το ΑΝ μέρος του κανόνα αναφέρεται σε ενέργειες των οποίων τα αποτελέσματα, αφού ο κανόνας εκτελεστεί, θα επιδράσουν, στην τρέχουσα κατάσταση του προβλήματος με τον ακόλουθο τρόπο: οι ενέργειες θα επιφέρουν αλλαγές στη μνήμη εργασίας και ιδιαίτερα στα γεγονότα που περιγράφουν τον κόσμο του προβλήματος είτε προσθέτοντας νέα γεγονότα είτε μεταβάλλοντας χαρακτηριστικά υπαρχόντων γεγονότων είτε διαγράφοντας υπάρχοντα γεγονότα, παράγοντας με τον τρόπο αυτό ένα νέο στιγμιότυπο του κόσμου του προβλήματος. Κανόνες με τέτοια χαρακτηριστικά καλούνται **κανόνες παραγωγής** ή **παραγωγικοί κανόνες** (production rules).

Παραδείγματα παραγωγικών κανόνων:

ΑΝ ο αγρότης βρίσκεται σε μια όχθη

ΤΟΤΕ μεταφέρε τον αγρότη στην απέναντι όχθη

ΑΝ το ζάρι φέρει 3

ΤΟΤΕ προχώρησε το πούλι 3 θέσεις μπροστά

Ουσιαστικά τα συστήματα παραγωγής είναι συστήματα κανόνων που περιέχουν τουλάχιστον έναν παραγωγικό κανόνα. Η βάση κανόνων τους καλείται **παραγωγική μνήμη** (production memory). Τα συστήματα παραγωγής ακολουθούν τη δομή και τον τρόπο λειτουργίας των συστημάτων κανόνων και δουλεύουν, όπως αυτοί, σε κύκλους.



Σχήμα 3.7 Δομή και λειτουργία Συστημάτων Παραγωγής

3.4.1 Κύκλοι Λειτουργίας

Σε κάθε κύκλο λειτουργίας ενός συστήματος κανόνων επιλέγεται ένας κανόνας από το σύνολο των κανόνων που μπορούν να πυροδοτηθούν και ως συνέπεια:

- νέοι ισχυρισμοί προστίθενται στη μνήμη εργασίας ως αποτέλεσμα της εκτέλεσης των παραγωγικών ενεργειών του μέρους ΤΟΤΕ του κανόνα και
- εκτελούνται οι καθορισμένες μη παραγωγικές ενέργειες του μέρους ΤΟΤΕ του κανόνα.

Ένας παραγωγικός κανόνας πρέπει να περιέχει τουλάχιστον μια παραγωγική ενέργεια και πιθανώς επιπλέον παραγωγικές και μη παραγωγικές ενέργειες. Η περιγραφή ενός κύκλου λειτουργίας σε βήματα είναι η ακόλουθη:

- **Βήμα 1.** Με τη βοήθεια του μηχανισμού ελέγχου της συμπερασματικής μηχανής γίνεται ταύτιση των ισχυρισμών κάθε κανόνα με τα γεγονότα (facts) που βρίσκονται στη μνήμη εργασίας (pattern matching).
- **Βήμα 2.** Οι κανόνες που έχουν αληθείς ισχυρισμούς (δηλαδή ισχυρισμούς που ταυτοποιήθηκαν με γεγονότα μέσα στη μνήμη εργασίας) συγκεντρώνονται μέσα σε ένα σύνολο που καλείται *σύνολο σύγκρουσης (conflict set)*.
- **Βήμα 3.** Αν το σύνολο σύγκρουσης είναι κενό ή ο μέγιστος αριθμός πυροδοτήσεων έχει συμπληρωθεί, τότε η εκτέλεση του συστήματος τερματίζεται.
- **Βήμα 4.** Πραγματοποιείται διάταξη των κανόνων μέσα στο σύνολο σύγκρουσης με βάση την ισχύουσα *στρατηγική επίλυσης συγκρούσεων (conflict resolution strategy)*.
- **Βήμα 5.** Ο πρώτος κανόνας του συνόλου σύγκρουσης πυροδοτείται επιφέροντας αλλαγές στα γεγονότα της μνήμης εργασίας.
- **Βήμα 6.** Επιστροφή στο Βήμα 1.



Σχήμα 3.8 Τα βήματα ενός κύκλου λειτουργίας

Δείτε [κινούμενη εικόνα 3.3 - Κύκλοι Λειτουργίας ενός Συστήματος Κανόνων](#)

3.4.2 Δομή Ελέγχου

Η δομή ελέγχου, σε κάθε κύκλο λειτουργίας, αναζητεί κανόνες των οποίων οι ισχυρισμοί είναι αληθείς, δηλαδή υπάρχουν ως γεγονότα στη μνήμη εργασίας. Οι κανόνες που έχουν εντοπιστεί να έχουν ικανοποιημένους ισχυρισμούς καταχωρίζονται σε ένα χώρο που καλείται *σύνολο σύγκρουσης* ή *ατζέντα (conflict set)*.

Σε ένα σύνολο σύγκρουσης:

- περιέχονται όλοι οι κανόνες των οποίων όλοι οι ισχυρισμοί είναι αληθείς, άρα είναι έτοιμοι προς πυροδότηση.
- εάν το IF μέρος ενός κανόνα περιέχει μία μεταβλητή, τότε ο κανόνας μπορεί να εμφανίζεται περισσότερες από μία φορές, μία για κάθε πιθανή ταυτοποίηση της μεταβλητής του με ένα διαφορετικό γεγονός της μνήμης εργασίας κατά τη διαδικασία ταιριάσματος προτύπων.
- εάν το IF μέρος περιέχει περισσότερες της μίας μεταβλητής, τότε ο κανόνας θα εμφανίζεται τόσες φορές όσοι και οι πιθανοί συνδυασμοί ταυτοποιήσεων μεταξύ των μεταβλητών και των γεγονότων της μνήμης εργασίας.

Η εκλογή ενός από τους κανόνες του συνόλου σύγκρουσης προς πυροδότηση καλείται **επίλυση σύγκρουσης** (conflict resolution) και γίνεται βάσει της ισχύουσας στρατηγικής επίλυσης συγκρούσεων. Μια τέτοια στρατηγική μπορεί να έχει διαφορετικά κριτήρια για την επιλογή κανόνα προς εκτέλεση, μερικά από τα οποία μπορεί να είναι:

- τυχαίος μηχανισμός επιλογής,
- προτεραιότητα που έχει δοθεί από το σχεδιαστή του συστήματος,
- επιλογή του πιο πρόσφατα ενημερωμένου κανόνα,
- επιλογή κανόνα με τους περισσότερους ισχυρισμούς,
- επιλογή κανόνα με τα περισσότερα συμπεράσματα.

Πρέπει να σημειωθεί ότι όλοι οι τρόποι επιλογής μπορεί να θεωρηθούν ευρετικοί, αφού δεν εγγυώνται την ορθότητα της επιλογής.

Βασικά θέματα σχεδιασμού συστημάτων εξαγωγής συμπερασμάτων είναι:

- ταίριασμα ισχυρισμών με γεγονότα ενεργού μνήμης (pattern matching),
- επιλογή κανόνα για πυροδότηση,
- χρονοδρομολόγηση κανόνων,
- εκτέλεση ενεργειών,
- τήρηση ιστορικού,
- εξασφάλιση μη ύπαρξης αμοιβαία αποκλειστικών ισχυρισμών,
- επιλογή δομής ελέγχου (forward vs backward chaining).

Η δομή ελέγχου μπορεί να βασίζεται στην προς τα εμπρός αλυσιδωτή εκτέλεση (forward chaining) ή στην προς τα πίσω αλυσιδωτή εκτέλεση (backward chaining) κανόνων. Για την εξαγωγή συμπερασμάτων στα συστήματα που ακολουθούν την προς τα εμπρός αλυσιδωτή πυροδότηση κανόνων χρησιμοποιείται το μοντέλο Modus Ponens, ενώ για αυτά που ακολουθούν την προς τα πίσω αλυσιδωτή πυροδότηση κανόνων χρησιμοποιείται το μοντέλο Modus Tollens (ανατρέξτε στην παράγραφο περί [συλλογιστικής](#)).

3.4.3 Προς τα εμπρός αλυσιδωτή εκτέλεση κανόνων

Η **προς τα εμπρός αλυσιδωτή εκτέλεση κανόνων** (forward chaining) είναι μια παραγωγική διαδικασία εξαγωγής συμπερασμάτων από κανόνες σύμφωνα με το μοντέλο έγκυρων συλλογισμών modus ponens, μίας από τις δυο μορφές έγκυρων συλλογισμών του προτασιακού λογισμού που χρησιμοποιεί την παραγωγή (deduction) για την εξαγωγή συμπερασμάτων.

Παράδειγμα συλλογισμού με modus ponens:

AN A TOTE B

Ισχύει το A επομένως ισχύει το B

Η προς τα εμπρός αλυσιδωτή εκτέλεση κανόνων πραγματοποιείται σε κύκλους λειτουργίας οδηγούμενη από τη συμπερασματική μηχανή και έχει τα παρακάτω χαρακτηριστικά:

- εκτέλεση οδηγούμενη από τα δεδομένα (data-driven),
- εκτέλεση από το βάθος προς την κορυφή του δένδρου των κανόνων (bottom-up),
- κίνηση προς τα εμπρός, δηλαδή από τους ισχυρισμούς προς τα συμπεράσματα ενός κανόνα.

Ο ψευδοκώδικας της προς τα εμπρός αλυσιδωτής εκτέλεσης είναι ο ακόλουθος:

ΟΣΟ το ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ όχι άδειο και το πρόβλημα δεν έχει λυθεί,
 Βρες όλους τους κανόνες που είναι έτοιμοι για πυροδότηση (όλοι οι ισχυρισμοί τους είναι αληθείς) και πρόσθεσέ τους στο ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ
 Εάν υπάρχουν ≥ 1 κανόνες στο ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ, διάλεξε έναν βάσει κάποιου κριτηρίου επιλογής (ΕΠΙΛΥΣΗ ΣΥΓΚΡΟΥΣΗΣ)
 Πυροδότησε τον κανόνα.
 Άλλαξε τη μνήμη εργασίας βάσει των αποτελεσμάτων της εκτέλεσης των ενεργειών του μέρους then του κανόνα.
 Τέλος ΟΣΟ

Παράδειγμα προς τα εμπρός αλυσιδωτής εκτέλεσης :

Στο παράδειγμά μας, στη μνήμη εργασίας βρίσκονται τα παρακάτω δυο γεγονότα:

Ελένη πληροφορικός
 Νίκος πληροφορικός

Επίσης, στη βάση γνώσης βρίσκονται οι παρακάτω δύο κανόνες:

R1 :
 IF ?X πληροφορικός
 THEN ?X είναι νοήμων

R2 :
 IF ?X είναι νοήμων
 THEN εμφάνισε "Μπράβο" ?X

Στον πρώτο κύκλο λειτουργίας του συστήματος, η συμπερασματική μηχανή θα πραγματοποιήσει ταίριασμα προτύπων (pattern matching) μεταξύ των γεγονότων της μνήμης εργασίας και των ισχυρισμών των κανόνων και θα εντοπίσει δύο κανόνες με ικανοποιημένους ισχυρισμούς τους οποίους και θα τοποθετήσει στο σύνολο σύγκρουσης.

Αυτοί είναι:

- α) ο κανόνας R1, όπου έχει ικανοποιηθεί ο ισχυρισμός ?X πληροφορικός, επειδή ταυτοποιήθηκε με το γεγονός Ελένη πληροφορικός, όπου πραγματοποιήθηκε η σύνδεση (binding) της μεταβλητής ?X με το Ελένη (?X = Ελένη)
- β) πάλι ο κανόνας R1, όπου έχει ικανοποιηθεί ο ισχυρισμός ?X πληροφορικός, επειδή ταυτοποιήθηκε με το γεγονός Νίκος πληροφορικός, όπου πραγματοποιήθηκε η σύνδεση (binding) της μεταβλητής ?X με το Νίκος (?X = Νίκος).

Ως αποτέλεσμα, το σύνολο σύγκρουσης θα διαμορφωθεί ως εξής:

Σύνολο σύγκρουσης: {R1 (?X=Ελένη) , R1 (?X=Νίκος) }

Στη συνέχεια θα επιχειρηθεί να επιλυθεί η σύγκρουση (conflict resolution) με την επιλογή ενός κανόνα από το σύνολο σύγκρουσης βάσει μιας στρατηγικής επίλυσης συγκρούσεων. Στο παράδειγμά μας, αυτή είναι η τυχαία επιλογή. Στον πρώτο κύκλο λειτουργίας, κατά την εφαρμογή του μηχανισμού τυχαίας επιλογής, επιλέγεται τυχαία προς πυροδότηση ο R1 (?X=Ελένη), ο οποίος και πυροδοτείται παράγοντας μέσα στη μνήμη εργασίας το νέο γεγονός "Ελένη είναι νοήμων" (βλέπε επόμενο πίνακα 3.1).

Παράδειγμα Forward Chaining

Κανόνας R1: IF ?X πληροφορικός THEN ?X είναι νοήμων		Κανόνας R2: IF ?X είναι νοήμων THEN εμφάνισε "Μπράβο" ?X	
Μνήμη Εργασίας	Σύνολο Σύγκρουσης	Κανόνας που πυροδοτείται	
Ελένη πληροφορικός Νίκος πληροφορικός	R1 (?X = Ελένη) R1 (?X = Νίκος)	Επιλέγεται τυχαία ο R1 (?X = Ελένη)	
Ελένη πληροφορικός Νίκος πληροφορικός Ελένη είναι νοήμων			

Πίνακας 3.1 Πρώτος κύκλος λειτουργίας παραδείγματος προς τα εμπρός αλυσιδωτής εκτέλεσης

Στη συνέχεια, παρουσιάζονται όλοι οι κύκλοι εκτέλεσης του συστήματος μέχρι την ολοκλήρωσή του, δηλαδή μέχρι να εξαχθούν όλα τα δυνατά συμπεράσματα και να εκτελεστούν οι πιθανές ενέργειες, έως ότου το σύνολο σύγκρουσης βρεθεί κενό:

1^{ος} Κύκλος

Μνήμη εργασίας (γεγονότα): Ελένη πληροφορικός, Νίκος πληροφορικός
 Σύνολο σύγκρουσης: {R1 (?X=Ελένη), R1 (?X=Νίκος)}
 Πυροδοτείται (τυχαία επιλογή) ο: R1 (?X = Ελένη)
 Συμπέρασμα (νέο γεγονός): Ελένη είναι νοήμων

2^{ος} Κύκλος

Μνήμη εργασίας (γεγονότα): Ελένη πληροφορικός, Νίκος πληροφορικός,
 Ελένη είναι νοήμων
 Σύνολο Σύγκρουσης: {R1 (?X=Νίκος), R2 (?X=Ελένη)}
 Πυροδοτείται (τυχαία επιλογή) ο: R2 (?X = Ελένη)
 Αποτέλεσμα: Μπράβο Ελένη

3^{ος} Κύκλος

Μνήμη εργασίας (γεγονότα): Ελένη πληροφορικός, Νίκος πληροφορικός,
 Ελένη είναι νοήμων
 Σύνολο σύγκρουσης: {R1 (?X=Νίκος)}
 Πυροδοτείται ο: R1 (?X=Νίκος)
 Συμπέρασμα (νέο γεγονός): Νίκος είναι νοήμων

4^{ος} Κύκλος

Μνήμη εργασίας (γεγονότα): Ελένη πληροφορικός, Νίκος πληροφορικός,

Ελένη είναι νοήμων, Νίκος είναι νοήμων
 Σύνολο σύγκρουσης : {R2 (?X=Νίκος) }
 Πυροδοτείται ο: R2 (?X = Νίκος)
 Αποτέλεσμα: Μπράβο Νίκος

Κατάσταση συστήματος κατά την έξοδο:

Μνήμη εργασίας (γεγονότα): Ελένη πληροφορικός, Νίκος πληροφορικός,
 Ελένη είναι νοήμων, Νίκος είναι νοήμων
 Σύνολο σύγκρουσης : { }
 Συμπεράσματα (νέα γεγονότα): Ελένη είναι νοήμων, Νίκος είναι νοήμων
 Αποτελέσματα: Μπράβο Ελένη, Μπράβο Νίκος

Μνήμη Εργασίας	Σύνολο Σύγκρουσης	Νέα Γεγονότα	Ενέργειες
Ελένη πληροφορικός Νίκος πληροφορικός Ελένη είναι νοήμων Νίκος είναι νοήμων	-	Ελένη είναι νοήμων Νίκος είναι νοήμων	Μπράβο Ελένη Μπράβο Νίκος

Πίνακας 3.2 Κατάσταση του συστήματος μετά τον τελευταίο κύκλο λειτουργίας παραδείγματος προς τα εμπρός αλυσιδωτής εκτέλεσης

Δείτε [κινούμενη εικόνα 3.4 - Λειτουργία της προς τα εμπρός αλυσιδωτής εκτέλεσης κανόνων](#)

Άλλο ενδιαφέρον παράδειγμα είναι αυτό του Ρομπότ (Βλαχάβας κ.ά, 2011).

Το πρόβλημα έχει ως εξής: Σε ένα πλαίσιο 10x10 θέσεων που περιέχει εμπόδια και αντικείμενα είναι τοποθετημένο ένα ρομπότ σε μια αρχική θέση (βλέπε [Σχήμα 1.9](#)) με μια συγκεκριμένη κατεύθυνση προς τα δυτικά, ανατολικά, βόρεια ή νότια. Το ρομπότ μπορεί να μετακινηθεί προς την κατεύθυνσή του σε μια γειτονική θέση, με την προϋπόθεση ότι η θέση αυτή δεν περιέχει εμπόδιο. Στόχος του προβλήματος είναι το ρομπότ να μετακινηθεί προς θέση, αποφεύγοντας τα εμπόδια, έως ότου βρεθεί σε κάποια θέση με αντικείμενο, οπότε να στείλει ένα μήνυμα και να σταματήσει.

Η μνήμη εργασίας για την αρχική κατάσταση του προβλήματος του ρομπότ, όπως αυτή αποτυπώνεται στο [σχήμα 1.9](#), είναι η ακόλουθη:

Μνήμη Εργασίας:

```
robot_at(6,4)
direction(e)
choice(w) choice(e) choice(n) choice(s)
obstacle_at(2,3) obstacle_at(3,8) obstacle_at(4,2)
obstacle_at(5,2) obstacle_at(5,6) obstacle_at(6,8)
obstacle_at(7,4) obstacle_at(7,7) obstacle_at(9,2)
object_at(4,4) object_at(4,7) object_at(7,2)
object_at(8,5) object_at(10,6)
```

όπου δηλώνουν:

- το γεγονός robot_at τη θέση του ρομπότ,
- το direction την κατεύθυνση προς την οποία κινείται το ρομπότ,

- τα 4 εναλλακτικά choice τις επιλογές που έχει το ρομπότ, αν θελήσει να αλλάξει κατεύθυνση (επιλογή θεωρείται και το να διατηρήσει την τρέχουσα κατεύθυνσή του),
- το obstacle_at τη θέση ενός εμποδίου,
- το object_at τη θέση ενός αντικειμένου.

Οι αντίστοιχοι κανόνες της βάσης γνώσης είναι οι ακόλουθοι:

- 1: **detect_object:** if robot_at(X,Y) and object_at(X,Y)
 then output('object is found'), exit
- 2: **move_west:** if robot_at(X,Y) and direction(w)
 then delwm(robot_at(X,Y)), NX=X-1,
 addwm(robot_at(NX,Y))
- 3: **move_east:** if robot_at(X,Y) and direction(e)
 then delwm(robot_at(X,Y)), NX=X+1
 addwm(robot_at(NX,Y))
- 4: **move_north:** if robot_at(X,Y) and direction(n)
 then delwm(robot_at(X,Y)), NY=Y+1
 addwm(robot_at(X,NY))
- 5: **move_south:** if robot_at(X,Y) and direction(s)
 then delwm(robot_at(X,Y)), NY=Y-1
 addwm(robot_at(X,NY))
- 6: **avoid_obstacle_south:** if robot_at(X,Y) and NY=Y-1
 and obstacle_at(X,NY) and direction(s)
 and choice(ND)
 then delwm(direction(s))
 addwm(direction(ND))
- 7: **avoid_obstacle_west:** if robot_at(X,Y) and NX=X-1
 and obstacle_at(NX,Y) and direction(w)
 and choice(ND)
 then delwm(direction(w))
 addwm(direction(ND))
- 8: **avoid_obstacle_north:** if robot_at(X,Y) and NY=Y+1
 and obstacle_at(X,NY) and direction(n)
 and choice(ND)
 then delwm(direction(n))
 addwm(direction(ND))
- 9: **avoid_obstacle_east:** if robot_at(X,Y) and NX=X+1
 and obstacle_at(NX,Y) and direction(e)
 and choice(ND)
 then delwm(direction(e))
 addwm(direction(ND))

Στον 1^ο κύκλο λειτουργίας, το σύνολο σύγκρουσης θα περιέχει 5 κανόνες, τον κανόνα 3 μετακίνησης ανατολικά **move_east** και τέσσερεις διαφορετικές εκδοχές του κανόνα 9 **avoid_obstacle_east**, μία για κάθε ταύτιση των ισχυρισμών του με τα γεγονότα robot_at(6 4) και choice(w), robot_at(6 4) και choice(e), robot_at(6 4) και choice(n) και, τέλος, robot_at(6 4) και choice(s) αντίστοιχα:

Σύνολο σύγκρουσης:

```
{3: move_east (X=6 και Y=4)
 9:avoid_obstacle_east (X=6 και Y=4 ND=w),
 9:avoid_obstacle_east (X=6 και Y=4 ND=n),
 9:avoid_obstacle_east (X=6 και Y=4 ND=s),
 9:avoid_obstacle_east (X=6 και Y=4 ND=e) }
```

Αν στον πρώτο κύκλο εκτέλεσης ως στρατηγική επίλυσης συγκρούσεων επιλεγεί η επιλογή του πιο ειδικού κανόνα (EE) και ως ειδικός κριθεί ο κανόνας **avoid_obstacle_east** έναντι του **move_east**, διότι η αποφυγή του εμποδίου έχει μεγαλύτερη σημασία από την κίνηση προς θέση με εμπόδιο, η επιλογή αυτή δε θα επιλύσει οριστικά τη σύγκρουση, δεδομένου ότι θα παραμένει προς επίλυση η σύγκρουση των τεσσάρων διαφορετικών εκδοχών του κανόνα **avoid_obstacle_east**. Εφόσον η επίλυση αφορά τέσσερις κανόνες με ίδια προτεραιότητα, επιβάλλεται η επιλογή νέας στρατηγικής για την επίλυση της σύγκρουσης. Θεωρούμε ότι ως νέα στρατηγική εφαρμόζεται αυτή της τυχαίας επιλογής (TE) που επιστρέφει τον κανόνα 9 **avoid_obstacle_east (ND=n)**, ο οποίος, όταν πυροδοτηθεί, θα διαγράψει από τη μνήμη εργασίας την τρέχουσα κατεύθυνση (`delwm(direction(e))`) και θα προσθέσει τη νέα (`addwm(direction(n))`).

Η λειτουργία του συστήματος ολοκληρώνεται, όταν το ρομπότ βρεθεί σε μία θέση όπου υπάρχει ένα αντικείμενο και πυροδοτηθεί ο κανόνας 1 **detect_object**. Οι κύκλοι λειτουργίας της εκτέλεσης του συστήματος παραγωγής που θα επιλύσει το πρόβλημα του ρομπότ παρουσιάζονται στον παρακάτω πίνακα.

Κύκλος	Μνήμη Εργασίας	Σύνολο Σύγκρουσης	Στρατηγική	Κανόνας που Πυροδοτείται
1	robot at (6,4) direction (e) choice (w) choice (e) choice (n) choice (s) obstacle_at (2,3) obstacle_at (3,8) ... object_at (4,4) object_at (4,7) ...	{3, 9 (ND=w), 9 (ND=n), 9 (ND=s), 9 (ND=e)}	EE (επιλογή πιο ειδικού) TE (τυχαία επιλογή)	9:avoid_obstacle_east (ND=n)
2	robot at (6,4) direction (n) ...	{4}	-	4:move_north
3	robot at (6,5) direction (n) ...	{4}	-	4:move_north
4	robot at (6,6) direction (n) ...	{4}	-	4:move_north
5	robot at (6,7) direction (n) ... obstacle_at (6,8) ...	{4, 8 (ND=w), 8 (ND=n), 8 (ND=s), 8 (ND=e)}	EE TE	8:avoid_obstacle_north (ND=n)
6	robot at (6,7) direction (n) ... obstacle_at (6,8) ...	{4, 8 (ND=w), 8 (ND=n), 8 (ND=s), 8 (ND=e)}	AE (αποφυγή επανάληψης) EE TE	8:avoid_obstacle_north (ND=n)
7	robot at (6,7) direction (e) ... obstacle_at (7,7) ...	{3, 9 (ND=w), 9 (ND=n), 9 (ND=s), 9 (ND=e)}	EE TE	9:avoid_obstacle_east (ND=w)
8	robot at (6,7) direction (w) ...	{2}	-	2:move_west
9	robot at (5,7) direction (w) ...	{2}	-	2:move_west
10	robot at (4,7) direction (w) object_at (4,7) ...	{1,2}	EE TE	1:detect_object

Πίνακας 3.3 Κύκλοι εκτέλεσης του προβλήματος του Ρομπότ

3.4.4 Προς τα πίσω αλυσιδωτή εκτέλεση κανόνων

Για να μπορέσει να εφαρμοστεί η **προς τα πίσω αλυσιδωτή εκτέλεση κανόνων** (backward chaining) απαιτείται να είναι από την αρχή γνωστός ο σκοπός της αναζήτησης.

Τα χαρακτηριστικά της προς τα πίσω αλυσιδωτής εκτέλεσης είναι:

- εκτελείται οδηγούμενη από το σκοπό (goal-driven).
- εκτελείται από την κορυφή προς το βάθος του δένδρου των κανόνων (top-down).
- εντοπίζεται στα συμπεράσματα των κανόνων.
- συχνά προλαβαίνει πυροδότηση κανόνων άσχετων προς το πρόβλημα προς λύση.

Περίληπτικά, ο τρόπος με τον οποίο λειτουργεί η προς τα πίσω αλυσιδωτή πυροδότηση των κανόνων είναι ο εξής: Αναζητούνται κανόνες που περιέχουν τον επιθυμητό στόχο στα συμπεράσματά τους. Αν βρεθούν ένας ή περισσότεροι μεταξύ αυτών με αληθείς ισχυρισμούς, συγκεντρώνονται στο σύνολο σύγκρουσης και πυροδοτείται αυτός που θα επιλεγεί με βάση την ισχύουσα στρατηγική επίλυσης σύγκρουσης, ολοκληρώνοντας με τον τρόπο αυτό την εκτέλεση του συστήματος.

Άλλως, αρχίζει ένας νέος κύκλος όπου σημειώνονται οι διαφορές του τρέχοντος κόσμου του προβλήματος από τον επιδιωκόμενο στόχο και κάθε διαφορά ορίζεται ως νέος υπο-στόχος. Στη συνέχεια, αναζητούνται κανόνες που μπορούν να ικανοποιήσουν τους υπο-στόχους και αυτοί που έχουν αληθείς ισχυρισμούς συγκεντρώνονται στο σύνολο σύγκρουσης. Ο μηχανισμός επίλυσης σύγκρουσης επιλέγει και πυροδοτεί έναν από αυτούς βάσει της ισχύουσας στρατηγικής και ο κύκλος ολοκληρώνεται με τη μείωση της διαφοράς που επέφερε ο κανόνας. Αν σε κάποιον κύκλο δε βρεθεί κανόνας που μειώνει τις εναπομείνουσες διαφορές, αυτό σημαίνει ότι δεν υπάρχει αλυσίδα κανόνων που οδηγεί στο στόχο, δηλαδή το πρόβλημα δεν έχει λύση.

Ψευδοκώδικας της προς τα πίσω αλυσιδωτής εκτέλεσης

Αν ο στόχος δεν έχει επιτευχθεί

 Θέσε ως ενδιάμεσους στόχους τις διαφορές μεταξύ του τρέχοντος κόσμου και το στόχο

ΟΣΟ υπάρχουν ενδιάμεσοι στόχοι που δεν έχουν επιτευχθεί

 Βρες όλους τους κανόνες που στο Then μέρος περιέχουν ενδιάμεσους στόχους και πρόσθεσέ τους στο ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ

 ΑΝ το ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ είναι άδειο

 ΤΟΤΕ εμφάνισε "Δεν υπάρχει αλυσίδα κανόνων που οδηγεί στο σκοπό μας" και ολοκλήρωσε τη διαδικασία αναζήτησης

 Αλλιώς

 ΑΝ υπάρχουν ≥ 1 κανόνες στο ΣΥΝΟΛΟ ΣΥΓΚΡΟΥΣΗΣ με αληθείς ισχυρισμούς

 ΤΟΤΕ επέλεξε έναν κανόνα βάσει κάποιου κριτηρίου επιλογής (ΕΠΙΛΥΣΗ ΣΥΓΚΡΟΥΣΗΣ) και πυροδότησέ τον

 Αλλιώς επέλεξε έναν κανόνα βάσει κάποιου κριτηρίου επιλογής και συμπλήρωσε τους μη ικανοποιημένους ισχυρισμούς του στους ενδιάμεσους στόχους

Τέλος ΟΣΟ

Επίστρεψε ως λύση τη λίστα των ικανοποιημένων ενδιάμεσων στόχων.

Παράδειγμα προς τα πίσω αλυσιδωτής εκτέλεσης κανόνα

Το πρόβλημα που έχουμε να λύσουμε με τη βοήθεια του συστήματός μας είναι η κατασκευή ενός κέικ. Τα απαραίτητα υλικά είναι το αλεύρι, τα αυγά, η ζάχαρη και το γάλα. Όταν υπάρχουν όλα, τότε η κατασκευή είναι δυνατή, αλλιώς πρέπει να αγοράσουμε όσα λείπουν πληρώνοντας ανάλογα χρήματα.

Σε κάθε κύκλο λειτουργίας του συστήματος, όσα υλικά είναι διαθέσιμα στην κουζίνα αποτελούν τα γεγονότα του προβλήματος μέσα στην τρέχουσα ενεργό μνήμη. Η γνώση σχετικά με το ποια υλικά είναι απαραίτητα για την κατασκευή του κέικ και πώς πρέπει να ενεργήσει κάποιος, όταν του λείπουν κάποια από αυτά, είναι η υπάρχουσα διαδικαστική γνώση για το πρόβλημα, η οποία αποτυπώνεται μέσα στους κανόνες. Στην αρχή του προβλήματος, στην κουζίνα υπάρχει μόνο ζάχαρη, αλλά υπάρχουν διαθέσιμα χρήματα – σε ρευστό και σε λογαριασμό μέσα στην τράπεζα. Βάσει αυτού, ως πρωταρχικός στόχος τίθεται η ικανοποίηση

των μη ικανοποιημένων ισχυρισμών του κανόνα που θα μας δώσει την κατασκευή του κέικ, του R1, όπως θα δούμε αμέσως μετά.

Μνήμη Εργασίας (γεγονότα)

Υπάρχει ζάχαρη
Υπάρχει χρήμα
Υπάρχει λογαριασμός στην τράπεζα

Κανόνες

R1 :

IF Υπάρχει αλεύρι AND Υπάρχει ζάχαρη AND Υπάρχει αυγά AND Υπάρχει γάλα
THEN εμφάνισε "Το κέικ μπορεί να κατασκευαστεί"

R2 :

IF Υπάρχει χρήμα AND NOT Υπάρχει αλεύρι
THEN ΠΡΟΣΘΕΣΕ Υπάρχει αλεύρι, ΑΦΑΙΡΕΣΕ Υπάρχει χρήμα

R3 :

IF Υπάρχει χρήμα AND NOT Υπάρχει ζάχαρη
THEN ΠΡΟΣΘΕΣΕ Υπάρχει ζάχαρη, ΑΦΑΙΡΕΣΕ Υπάρχει χρήμα

R4 :

IF Υπάρχει χρήμα AND NOT Υπάρχουν αυγά
THEN ΠΡΟΣΘΕΣΕ Υπάρχουν αυγά, ΑΦΑΙΡΕΣΕ Υπάρχει χρήμα

R5 :

IF Υπάρχει χρήμα AND NOT Υπάρχει γάλα
THEN ΠΡΟΣΘΕΣΕ Υπάρχει γάλα, ΑΦΑΙΡΕΣΕ Υπάρχει χρήμα

R6 :

IF Υπάρχει λογαριασμός στην τράπεζα AND NOT Υπάρχει χρήμα
THEN ΠΡΟΣΘΕΣΕ Υπάρχει χρήμα

Στόχος (ικανοποίηση των ανικανοποίητων ισχυρισμών του R1) η απόκτηση των:

Υπάρχει αλεύρι, Υπάρχουν αυγά, Υπάρχει γάλα

Το ζητούμενο του προβλήματος, δηλαδή η δυνατότητα κατασκευής κέικ, θα επιτευχθεί, όταν εκτελεστεί ο κανόνας R1, δηλαδή όταν ικανοποιηθούν όλοι οι ισχυρισμοί του. Σε κάθε κύκλο εργασίας, στο σύνολο σύγκρουσης συγκεντρώνονται οι κανόνες που περιέχουν στα συμπεράσματά τους τουλάχιστον ένα γεγονός που ταυτίζεται με ανικανοποίητο ισχυρισμό. Η εκτέλεση σε κύκλους, μέχρι να βρεθεί το ζητούμενο, έχει ως εξής:

1^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει χρήμα, Υπάρχει λογαριασμός στην τράπεζα

Ενδιάμεσος στόχος: Υπάρχει αλεύρι, Υπάρχουν αυγά, Υπάρχει γάλα

Σύνολο σύγκρουσης :

R2 (δίνει ως συμπέρασμα το Υπάρχει αλεύρι)

R4 (δίνει ως συμπέρασμα το Υπάρχουν αυγά)

R5 (δίνει ως συμπέρασμα το Υπάρχει γάλα)

Επιλέγεται ο R2 και πυροδοτείται.

Νέο γεγονός: Υπάρχει αλεύρι

2^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει αλεύρι, Υπάρχει λογαριασμός στην τράπεζα

Ενδιάμεσος στόχος: Υπάρχουν αυγά, Υπάρχει γάλα

Σύνολο σύγκρουσης :

R4 (δίνει ως συμπέρασμα το Υπάρχουν αυγά)

R5 (δίνει ως συμπέρασμα το Υπάρχει γάλα)

Επιλέγεται ο R4, αλλά έχει μη ικανοποιημένους ισχυρισμούς και δεν μπορεί να πυροδοτηθεί.

Νέος ενδιάμεσος στόχος ο ανικανοποίητος ισχυρισμός του R4: Υπάρχει χρήμα

Νέο σύνολο σύγκρουσης :

R6 (δίνει ως συμπέρασμα το Υπάρχει χρήμα), R4, R5

Επιλέγεται ο R6 και πυροδοτείται.

Νέο γεγονός στη μνήμη εργασίας: Υπάρχει χρήμα

3^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει αλεύρι, Υπάρχει χρήμα, Υπάρχει λογαριασμός στην τράπεζα

Ενδιάμεσος στόχος: Υπάρχουν αυγά, Υπάρχει γάλα

Σύνολο σύγκρουσης :

R4 (δίνει ως συμπέρασμα το Υπάρχουν αυγά)

R5 (δίνει ως συμπέρασμα το Υπάρχει γάλα)

Επιλέγεται ο R4 και πυροδοτείται.

Νέο γεγονός στη μνήμη εργασίας: Υπάρχουν αυγά

4^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει αλεύρι, Υπάρχουν αυγά, Υπάρχει λογαριασμός στην τράπεζα

Ενδιάμεσος στόχος: Υπάρχει γάλα

Σύνολο σύγκρουσης :

R5 (δίνει ως συμπέρασμα το Υπάρχει γάλα)

Επιλέγεται ο R5, αλλά έχει μη ικανοποιημένους ισχυρισμούς και δεν μπορεί να πυροδοτηθεί.

Νέος ενδιάμεσος στόχος ο ανικανοποίητος ισχυρισμός του R5: Υπάρχει χρήμα

Νέο σύνολο σύγκρουσης :

R6 (δίνει ως συμπέρασμα το Υπάρχει χρήμα), R5

Επιλέγεται ο R6 και πυροδοτείται.

Νέο γεγονός στη μνήμη εργασίας: Υπάρχει χρήμα

5^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει αλεύρι, Υπάρχουν αυγά, Υπάρχει λογαριασμός στην τράπεζα, Υπάρχει χρήμα

Ενδιάμεσος στόχος: Υπάρχει γάλα

Σύνολο σύγκρουσης :

R5 (δίνει ως συμπέρασμα το Υπάρχει γάλα)

Επιλέγεται ο R5 και πυροδοτείται.

Νέο γεγονός στη μνήμη εργασίας: Υπάρχει γάλα

6^{ος} Κύκλος

Μνήμη εργασίας: Υπάρχει ζάχαρη, Υπάρχει αλεύρι, Υπάρχουν αυγά,
Υπάρχει λογαριασμός στην τράπεζα, υπάρχει γάλα

Ενδιάμεσος στόχος: -

Σύνολο σύγκρουσης :

R1 (στόχος της αναζήτησης)

Πυροδοτείται ο R1

Εμφανίζεται: "Το κέικ μπορεί να κατασκευαστεί"

Δείτε [κινούμενη εικόνα 3.5 - Λειτουργία της προς τα πίσω αλυσιδωτής εκτέλεσης κανόνων](#)

3.4.5 Ο Γενικός Επιλυτής Προβλημάτων (General Problem Solver)

Ο αλγόριθμος της προς τα πίσω αλυσιδωτής εκτέλεσης βασίζεται στη θεωρία των Newell & Simon (1972) σχετικά με το χώρο επίλυσης προβλημάτων, η οποία αναπτύχθηκε στο πλαίσιο της Γνωστικής Επιστήμης και που συνοπτικά αναφέρει τα εξής:

Η ανθρώπινη συμπεριφορά στην επίλυση προβλημάτων μπορεί να θεωρηθεί ως η δημιουργία καταστάσεων γνώσης, με εφαρμογή νοητικών τελεστών και μετάβαση από μια αρχική κατάσταση σε μία τελική. Οι τελεστές αυτοί υπαγορεύουν τις επιτρεπόμενες κινήσεις στο χώρο ή ποιες κινήσεις απαγορεύονται, όταν κάποιες ειδικές συνθήκες ισχύουν.

Για κάθε δεδομένο πρόβλημα υπάρχει ένα μεγάλο πλήθος εναλλακτικών δρόμων που οδηγούν στο στόχο του. Το σύνολο των καταστάσεων που δημιουργούν οι τελεστές κατά την αναζήτηση λύσης αποκαλείται **βασικός χώρος επίλυσης προβλήματος** (basic problem solution space).

Οι άνθρωποι χρησιμοποιούν τις γνώσεις τους και διάφορες ευρετικές μεθόδους, για να δημιουργήσουν το βασικό χώρο επίλυσης των προβλημάτων και να αναζητήσουν τη συντομότερη λύση.

Όλα αυτά λειτουργούν με τους περιορισμούς του ανθρώπινου γνωστικού συστήματος, δηλαδή λαμβανομένων υπόψη των ορίων του χώρου της μνήμης μικρής διάρκειας και των ορίων στην ταχύτητα με την οποία μπορούν να μεταφέρονται οι πληροφορίες από και προς τη μνήμη μακράς διάρκειας.

Ήδη το 1959 οι προαναφερθέντες ερευνητές είχαν δημιουργήσει την πρώτη εκδοχή του **Γενικού Επιλυτή Προβλημάτων - ΓΕΠ** (General Problem Solver - GPS) ως γενικού συστήματος επίλυσης προβλημάτων (Newell κ.ά., 1959).

Ο ΓΕΠ/GPS είναι ένας μηχανισμός για επίλυση προβλημάτων μέσω ηλεκτρονικού υπολογιστή που αποτελεί μοντέλο του ανθρώπινου τρόπου επίλυσης προβλημάτων με βάση τη συλλογιστική. Το σύστημα χρησιμοποιεί στοίβα αποθήκευσης ενδιάμεσων στόχων και βασίζεται στην Means-Ends Analysis (MEA), μια μέθοδο επίλυσης προβλημάτων στο χώρο της ΤΝ για τον περιορισμό του χώρου αναζήτησης .

Η MEA υλοποιεί τη θεωρία των ερευνητών για το χώρο επίλυσης προβλημάτων με τα εξής βήματα:

1. Σημείωσε τις διαφορές μεταξύ της τρέχουσας κατάστασης και του στόχου.
2. Δημιούργησε έναν ενδιάμεσο στόχο, για να μειώσεις αυτές τις διαφορές.
3. Διάλεξε έναν τελεστή που εφαρμοζόμενος θα ικανοποιήσει τον τεθέντα ενδιάμεσο στόχο.
4. Αν ο τελεστής έχει ανικανοποίητες προϋποθέσεις, για να εφαρμοστεί, θέσε μία από αυτές τις προϋποθέσεις ως νέο ενδιάμεσο στόχο και συνέχισε.

Ο αλγόριθμος του ΓΕΠ/GPS είναι ο εξής:

Τοποθέτησε τελικό σκοπό στη ΣΤΟΙΒΑ
ΟΣΟ η στοίβα όχι κενή και όχι κορεσμένη

Πρόσθεσε τις διαφορές μεταξύ του τρέχοντα στόχου και του υπάρχοντα κόσμου στη στοίβα
Βρες όλους τους κανόνες των οποίων τα συμπεράσματα περιέχουν τον τρέχοντα στόχο και βάλε τους στο σύνολο σύγκρουσης
ΕΑΝ υπάρχουν ένας ή περισσότεροι κανόνες στο σύνολο σύγκρουσης
ΤΟΤΕ διάλεξε έναν
 ΕΑΝ ο κανόνας είναι έτοιμος για πυροδότηση
 ΤΟΤΕ πυροδότησέ τον
 ΑΛΛΙΩΣ πρόσθεσε τις μη ικανοποιημένες προϋποθέσεις στη στοίβα των στόχων
ΑΛΛΙΩΣ δεν υπάρχει κανόνας που λύνει το πρόβλημα
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

Κλασικό παράδειγμα κατάλληλο, για να παρουσιάσει τον τρόπο υλοποίησης του ΓΕΠ/GPS, είναι η επίλυση του προβλήματος των πύργων του Ανόι με αναδρομή.

Γενικά, η επίλυση ενός προβλήματος με συλλογιστική που βασίζεται στην προς τα πίσω αλυσιδωτή πυροδότηση κανόνων μπορεί να προσομοιωθεί με τον τρόπο συλλογισμού ενός γιατρού, ο οποίος προσπαθώντας να επιβεβαιώσει τις υποψίες του για την αιτία μιας ασθένειας κάνει στον ασθενή συγκεκριμένες ερωτήσεις για τα συμπτώματα που θα επιβεβαιώσουν την υπόθεσή του. Ένας επιστήμων στα Μαθηματικά σκέφτεται πολλές φορές με παρόμοιο τρόπο, για να καταλήξει στην απόδειξη ενός θεωρήματος που ισχυρίζεται ότι ισχύει.

Στο [κεφάλαιο περί εμπειρών συστημάτων](#) θα αναλύσουμε βαθύτερα τις μεθόδους πυροδότησης κανόνων καθώς και τις στρατηγικές επίλυσης συγκρούσεων σε καθεμία, θα συγκρίνουμε τα προτερήματα και μειονεκτήματά τους στο πλαίσιο της επίλυσης προβλημάτων του πραγματικού κόσμου και θα παρουσιάσουμε το τεχνολογικό πλαίσιο μέσα στο οποίο πρέπει να σχεδιάζονται.

Βιβλιογραφία/Αναφορές

- Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems* (2^η έκδοση). Αγγλία: Addison-Wesley.
- Newell, A., Shaw, J.C. & Simon, H.A. (1959). Report on a general problem-solving program. *UNESCO Proc. the Int. Conf. on Information Processing ICIP 1.6.8.* (pp. 256–264). Παρίσι: Unesco House.
Ανακτήθηκε από http://bitsavers.informatik.uni-stuttgart.de/pdf/rand/ipl/P-1584_Report_On_A_General_Problem-Solving_Program_Feb59.pdf
- Newell, A. & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, Νιου Τζέρσεϊ: Prentice-Hall.
- Russell, S. J. & Norvig, P. (2007). *Τεχνητή Νοημοσύνη: μια σύγχρονη προσέγγιση*. Αθήνα:Κλειδάριθμος.
- Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κόκκορας, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστημίου Μακεδονίας.
- Δουκίδης, Γ., & Αγγελίδης, Μ. (1998). *Εμπειρα συστήματα, Τεχνητή Νοημοσύνη και Lisp*. Αθήνα: Ι. Σιδέρης.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Αντιστοιχίστε σωστά τις μεθόδους συλλογιστικής με τους ορισμούς που δίνονται

1. Επαγωγή	A. Χρήση κανόνων για εξαγωγή συμπερασμάτων
2. Παραγωγή	B. Εξαγωγή συμπερασμάτων από υποδειγματικές περιπτώσεις
3. Παραδρομή	C. Υιοθέτηση της «καλύτερης» περίπτωσης για εξαγωγή συγκεκριμένου συμπεράσματος
4. Απαγωγή	D. Δεν αποτελεί μέθοδο συλλογισμού.

Απάντηση/Λύση

1 → B, 2 → A, 3 → D, 4 → C

Κριτήριο αξιολόγησης 2

Απαντήστε ποια από τα παρακάτω αποτελούν έγκυρα συμπεράσματα του κανόνα που ακολουθεί:

ΑΝ κάποιος μελετά σκληρά, ΤΟΤΕ γράφει καλά στις εξετάσεις.

1.	Η Μαρία μελετά σκληρά· άρα, θα γράψει καλά στις εξετάσεις.
2.	Η Μαρία δε μελετά σκληρά· άρα, δε θα γράψει καλά στις εξετάσεις.
3.	Η Μαρία έγραψε καλά στις εξετάσεις· άρα, μελετά σκληρά.
4.	Η Μαρία δεν έγραψε καλά στις εξετάσεις· άρα, δε μελετά σκληρά.

Απάντηση/Λύση

Έγκυρα συμπεράσματα είναι το 1, γιατί ακολουθεί το έγκυρο μοντέλο συλλογισμού Modus Ponens, και το 4, γιατί ακολουθεί το έγκυρο μοντέλο συλλογισμού Modus Tollens.

Κριτήριο αξιολόγησης 3

Ποια από τα παρακάτω κάνει η δομή ελέγχου μιας συμπερασματικής μηχανής;

- A. Όταν πυροδοτείται ένας κανόνας, τοποθετεί τα νέα γεγονότα που προκύπτουν από την εκτέλεση των ενεργειών του μέρους then στην ενεργό μνήμη.
- B. Όταν ξεκινά ένας νέος κύκλος λειτουργίας, προσπαθεί να διαπιστώσει αν ικανοποιούνται όλοι οι ισχυρισμοί ενός κανόνα, ώστε να τον τοποθετήσει στο σύνολο σύγκρουσης.
- Γ. Σε κάθε κύκλο λειτουργίας δεσμεύει τις μεταβλητές που εμφανίζονται στους ισχυρισμούς κάθε κανόνα με ιδιότητες αντίστοιχων γεγονότων στην ενεργό μνήμη.

Απάντηση/Λύση

Το σωστό είναι το B.

Κριτήριο αξιολόγησης 4

Μπορεί το σύνολο σύγκρουσης να χαρακτηριστεί ως ο χώρος όπου συγκεντρώνονται όλοι οι κανόνες που πρόκειται να πυροδοτηθούν;

ΝΑΙ/ΟΧΙ

Απάντηση/Λύση

ΟΧΙ. Συγκεντρώνονται όλοι οι κανόνες που θα μπορούσαν να πυροδοτηθούν. Από αυτούς μόνο ένας θα επιλεγεί για πυροδότηση, αφού επιλυθεί η σύγκρουση βάσει μιας στρατηγικής επίλυσης συγκρούσεων.

Κριτήριο αξιολόγησης 5

Πότε ολοκληρώνονται οι κύκλοι λειτουργίας σε ένα σύστημα παραγωγής;

- A. Όταν το σύστημα συμπληρώσει έναν προκαθορισμένο αριθμό κύκλων λειτουργίας.
- B. Όταν δεν υπάρχουν κανόνες προς πυροδότηση μέσα στο σύνολο σύγκρουσης.
- Γ. Όταν τρέξει ο κανόνας που εντοπίζει το στόχο.

Απάντηση/Λύση

Η ορθή απάντηση είναι το B, όταν δεν υπάρχουν κανόνες προς πυροδότηση μέσα στο σύνολο σύγκρουσης. Το ότι τρέχει ο κανόνας που εντοπίζει το στόχο δε σημαίνει ότι αν βρεθεί στη συνέχεια και άλλος κανόνας στο σύνολο σύγκρουσης, δε θα εκτελεστεί στον επόμενο κύκλο. Προκαθορισμένος αριθμός κύκλων λειτουργίας δε μπορεί να οριστεί παραμετρικά.

Κριτήριο αξιολόγησης 6

Σε ένα σύστημα παραγωγής, η υπάρχουσα εμπειρία για το προς επίλυση πρόβλημα βρίσκεται αποτυπωμένη:

- A. στη μνήμη εργασίας,
- B. στη βάση γνώσης,
- Γ. στο σύνολο σύγκρουσης,
- Δ. σε όλα τα προηγούμενα.

Απάντηση/Λύση

Η ορθή απάντηση είναι το B, γιατί η υπάρχουσα εμπειρία υπό μορφή κανόνων αποτελεί τη γνώση για το πρόβλημα που σε κάθε παραγωγικό σύστημα είναι αποτυπωμένη μέσα στη βάση γνώσης. Η μνήμη εργασίας δεν περιέχει γνώση, αλλά γεγονότα που περιγράφουν ένα στιγμιότυπο του κόσμου του προβλήματος. Το σύνολο σύγκρουσης δεν περιέχει τη γνώση για το πρόβλημα· απλώς, είναι ένας βοηθητικός χώρος της λειτουργίας του συστήματος όπου σε κάθε κύκλο του συγκεντρώνονται οι κανόνες που έχουν αληθείς ισχυρισμούς.

ΚΕΦΑΛΑΙΟ 4 –Μηχανική Μάθηση

Σύνοψη

Η επίλυση προβλημάτων με ευφυείς προσεγγίσεις βασίζεται στη δυνατότητα των συστημάτων να μαθαίνουν. Στο κεφάλαιο αυτό σκιαγραφείται η επιστημονική περιοχή της Μηχανικής Μάθησης (Machine Learning) και ιδιαίτερα αυτή της Επαγωγικής Μάθησης (Inductive Learning). Στο πλαίσιο αυτό, παρουσιάζονται αλγόριθμοι που υποστηρίζουν την Επιβλεπόμενη Επαγωγική Μάθηση με παραδείγματα, όπως τα Δένδρα Απόφασης (Decision Trees), πρακτικές Επαγωγικής Μάθησης Μέσω Συνάφειας με Περιπτώσεις, όπως η Συλλογιστική Βασισμένη σε Περιπτώσεις (Case-Based Reasoning), συστήματα Μη Συμβολικής ΤΝ, όπως τα Νευρωνικά Δίκτυα (Neural Networks), αλγόριθμοι Εξελικτικής Μάθησης, όπως οι Γενετικοί Αλγόριθμοι (Genetic Algorithms).

Προαπαιτούμενη γνώση

Αναπαράσταση γνώσης, επίλυση προβλημάτων.

4.1 Εισαγωγή

Η **Μάθηση** (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Παρά τις μελέτες και τις έρευνες επί χρόνια από τους επιστήμονες του πεδίου της Γνωστικής Ψυχολογίας και τους φιλοσόφους, η έννοια της μάθησης δεν έχει γίνει πλήρως κατανοητή. Πώς, λοιπόν, θα μπορούσαν οι επιστήμονες του χώρου της ΤΝ να δημιουργήσουν υπολογιστικά συστήματα ικανά να μάθουν, να επιτύχουν, δηλαδή, τη λεγόμενη **Μηχανική Μάθηση** (Machine Learning).

Αυτή μπορεί να οριστεί ως:

το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου.

Βάσει του ορισμού αυτού, η Μηχανική Μάθηση έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας. Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης δίνεται από τον Mitchell (1997):

«Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετρείται από το P , βελτιώνεται μέσω της εμπειρίας E .»

Στην **Επαγωγική Μάθηση** (Inductive Learning), με τη διαδικασία της **επαγωγής** (induction) ο άνθρωπος μαθαίνει κατανοώντας το περιβάλλον του μέσω παρατηρήσεων και δημιουργεί μια απλοποιημένη (αφαιρετική) εκδοχή του που ονομάζεται **νοητικό μοντέλο** (mental model). Επιπλέον, ο άνθρωπος έχει τη δυνατότητα να οργανώνει και να συσχετίζει τις εμπειρίες και τις παρατηρήσεις του δημιουργώντας νέες δομές που ονομάζονται **νοητικά πρότυπα** (mental patterns), με αξιοποίηση και του επαγωγικού και του απαγωγικού συλλογισμού. Στη δημιουργία νέων προτύπων από παλαιά βασίζονται οι τρόποι μάθησης που εξαρτώνται σε μεγαλύτερο ή μικρότερο βαθμό από την προϋπάρχουσα γνώση για ένα πρόβλημα, όπως είναι η μάθηση από επεξηγήσεις και η μάθηση από περιπτώσεις.

Σε σχέση με την ανθρώπινη ικανότητα προς μάθηση, οι φιλόσοφοι θέτουν το ερώτημα: «Πώς μπορεί ένας επαγωγικός συλλογισμός που οδηγεί στη μάθηση να αξιολογηθεί ως προς την ορθότητά του;». Αντίστοιχα, οι ψυχολόγοι ρωτούν: «Πώς αποθηκεύει ο εγκέφαλος τα αποτελέσματα της διαδικασίας της μάθησης, δηλαδή τα νοητικά μοντέλα και τα πρότυπα;». Στο χώρο της ΤΝ απλώς ρωτούν: «Πώς μπορεί μία μηχανή να δημιουργήσει νέα μοντέλα και πρότυπα μάθησης από συγκεκριμένα παραδείγματα και πόσο αξιόπιστα είναι αυτά τα μοντέλα και πρότυπα στην πράξη;».

Με βάση τα παραπάνω, μπορεί να δοθεί ο ακόλουθος εναλλακτικός ορισμός για τη Μηχανική Μάθηση:

Μηχανική Μάθηση ονομάζεται η ικανότητα ενός υπολογιστικού συστήματος να δημιουργεί μοντέλα ή πρότυπα από ένα σύνολο δεδομένων.

Ως κλάδος της ΤΝ, η Μηχανική Μάθηση ασχολείται με τη μελέτη αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους.

Όσον αφορά τη σχεδίαση των συστημάτων Μηχανικής Μάθησης, για τα συστήματα που ανήκουν στη συμβολική ΤΝ, η δυνατότητα μάθησης προσδιορίζεται ως η ικανότητα πρόσκτησης επιπλέον γνώσης, που επιφέρει μεταβολές στην υπάρχουσα καταχωρημένη γνώση είτε αλλάζοντας χαρακτηριστικά της είτε με αυξομείωσή της. Στην περίπτωση των συστημάτων ΤΝ που ανήκουν στη Μη Συμβολική ΤΝ (όπως η περίπτωση των Τεχνητών Νευρωνικών Δικτύων), ως μάθηση προσδιορίζεται η δυνατότητα που διαθέτουν τα συστήματα στο να μετασχηματίζουν την εσωτερική τους δομή, παρά στο να μεταβάλλουν κατάλληλα τη γνώση που έχει καταχωρηθεί μέσα σε αυτά κατά το σχεδιασμό τους.

Αν και απέχουμε πάρα πολύ από τη δημιουργία μηχανών που μαθαίνουν τόσο καλά όσο ο άνθρωπος, για συγκεκριμένες περιοχές μάθησης έχουν αναπτυχθεί αλγόριθμοι οι οποίοι έχουν επιτρέψει την εμφάνιση σύγχρονων εμπορικών εφαρμογών με σημαντική επιτυχία. Επιπλέον, τα αποτελέσματα από τις εφαρμογές της ΤΝ αρχίζουν ήδη να είναι ορατά και να δίνουν απαντήσεις σε αναπάντητα, έως τώρα, ερωτήματα των άλλων κλάδων που διερευνούν την ικανότητα του ανθρώπου να μαθαίνει.

Ο τομέας της Μηχανικής Μάθησης αναπτύσσει, επίσης, επιτυχώς την **Εξελικτική Μάθηση** (Evolutionary Learning), η οποία μιμείται διαδικασίες φυσικής αναπαραγωγικής σε έμβια όντα. Χρησιμοποιείται κυρίως σε προβλήματα βελτιστοποίησης. Στην Εξελικτική Μάθηση κυριαρχούν οι γενετικοί αλγόριθμοι που θα παρουσιαστούν στο τέλος του κεφαλαίου.

Εκτός της ίδιας της ΤΝ, μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της Μηχανικής Μάθησης συγκαταλέγονται οι: Εξόρυξη Δεδομένων, Πιθανότητες και Στατιστική, Θεωρία της Πληροφορίας, Αριθμητική Βελτιστοποίηση, Θεωρία της Πολυπλοκότητας, Θεωρία Ελέγχου (προσαρμοστική), Ψυχολογία (εξελικτική, γνωστική), Νευροβιολογία και Γλωσσολογία.

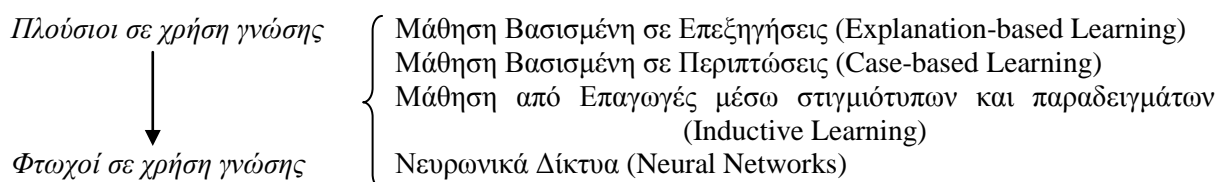
4.2 Είδη Μηχανικής Μάθησης

Εν γένει, ο τομέας της Μηχανικής Μάθησης αναπτύσσει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση. Πιο αναλυτικά:

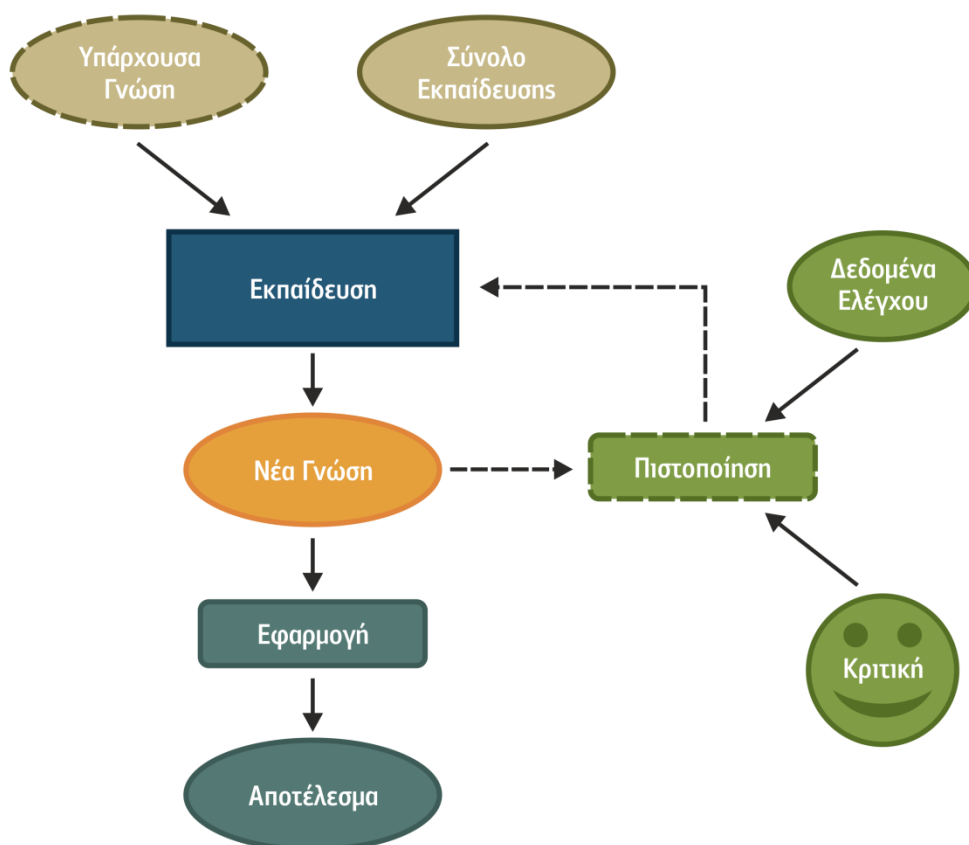
- **Επιβλεπόμενη Μάθηση** (Supervised Learning) είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Χρησιμοποιείται σε προβλήματα:
 - Ταξινόμησης (Classification)
 - Πρόγνωσης (Prediction)
 - Διερμηνείας (Interpretation)
- **Μη Επιβλεπόμενη Μάθηση** (Unsupervised Learning), όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Χρησιμοποιείται σε προβλήματα:
 - Ανάλυσης Συσχετισμών (Association Analysis)
 - Ομαδοποίησης (Clustering)
- **Ενισχυτική Μάθηση** (Reinforcement Learning), όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

Για κάθε πρόβλημα προς επίλυση στο χώρο της Μηχανικής Μάθησης υπάρχει ένας κατάλληλος τρόπος μάθησης και για κάθε τρόπο μάθησης υπάρχει τουλάχιστον ένας κατάλληλος αλγόριθμος που μπορεί να χρησιμοποιηθεί.

Όλοι οι αλγόριθμοι Μηχανικής Μάθησης διαχειρίζονται τη γνώση αναπαριστώντας την με κάποιον από τους τρόπους που έχουμε ήδη παρουσιάσει στο [κεφάλαιο 2](#) για την αναπαράσταση της γνώσης ή με άλλους πιο μαθηματικοποιημένους τρόπους που θεωρούνται από τον συγκεκριμένο αλγόριθμο καταλληλότεροι να την εκφράσουν. Ορισμένοι αλγόριθμοι δέχονται ως είσοδο μόνο παρατηρήσεις και άλλοι λαμβάνουν υπόψη τους λίγο ή περισσότερο την προϋπάρχουσα γνώση. Μια προσπάθεια κατάταξης των αλγορίθμων με κριτήριο τον τρόπο μάθησης βασισμένο περισσότερο ή λιγότερο στην υπάρχουσα γνώση δίνεται παρακάτω:



Στο σχήμα 4.1, αποτυπώνεται ο γενικός τρόπος λειτουργίας των αλγορίθμων Μηχανικής Μάθησης. Η βασικότερη φάση κάθε αλγόριθμου είναι η εκπαίδευση, όπου ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα **σύνολο δεδομένων εκπαίδευσης** (training set) προς επίτευξη του σκοπού του, τη δημιουργία νέας γνώσης. Επιπλέον, μπορεί είτε να χρησιμοποιήσει λιγότερο ή περισσότερο την υπάρχουσα γνώση είτε να μην τη χρησιμοποιήσει καθόλου.



Σχήμα 4.1 Φάσεις Μηχανικής Μάθησης

Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης της παραγόμενης νέας γνώσης. Συνήθως, η πιστοποίηση πραγματοποιείται καταρχάς από τον ίδιο τον αλγόριθμο μέσω διαδικασιών ανάκλησης (recall) με τη βοήθεια **δεδομένων ελέγχου** (test data) και, στη συνέχεια, μέσω κριτικής που κάνει ο χρήστης βάσει των γνώσεων που διαθέτει για το πρόβλημα που επιχειρεί να λύσει ο αλγόριθμος. Τέλος, η νέα γνώση δίνεται προς χρήση σε εφαρμογές στις οποίες είναι απαραίτητη, για να λυθούν πραγματικά προβλήματα.

4.3 Αλγόριθμοι Επιβλεπόμενης Επαγωγικής Μάθησης

Ένας από τους συνηθέστερους τρόπους Μηχανικής Μάθησης είναι μέσω επαγωγών οι οποίες παρέχουν δια του προτασιακού λογισμού ένα μηχανισμό εξαγωγής (όχι απαραίτητα σωστών) λογικών συμπερασμάτων από παραδείγματα.

Οι **αλγόριθμοι Επαγωγικής Μάθησης** (Inductive Learning algorithms) είναι ένα είδος αλγορίθμων που αναπτύχθηκε στο πλαίσιο της ΤΝ και ιδιαίτερα στο χώρο της Μηχανικής Μάθησης. Σκοπός των αλγορίθμων αυτών είναι η κατάληξη σε αποφάσεις σχετικές με τις σχέσεις που κυριαρχούν μέσα σε ένα σύνολο παραδειγμάτων που έχουν συγκεντρωθεί από παρατηρήσεις. Οι αλγόριθμοι επιβλεπόμενης Επαγωγικής Μάθησης εφαρμόζονται κυρίως σε προβλήματα ταξινόμησης (classification problems) και σε προβλήματα παρεμβολής (regression problems). Δημιουργούνται μοντέλα πρόβλεψης διακριτών τάξεων κατά την κατηγοριοποίηση και αριθμητικών τιμών κατά την παρεμβολή. Για προβλήματα πρόβλεψης, χρησιμοποιείται κυρίως μια παραλλαγή τους, οι αλγόριθμοι ημι-επιβλεπόμενης Μηχανικής Μάθησης, οι οποίοι λειτουργούν με σύνολο εκπαίδευσης μέσα στο οποίο υπάρχουν παραδείγματα μη γνωστές εξόδους.

Στην **επιβλεπόμενη Επαγωγική Μάθηση** (supervised Inductive Learning) το σύστημα πρέπει να “μάθει”, δηλαδή να κατασκευάσει ένα νέο μοντέλο υπό μορφή μιας **συνάρτησης πρόγνωσης** (predictor function), η οποία θα απεικονίζει δεδομένες εισόδους σε γνωστές, επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Για τη συνάρτηση πρόγνωσης ισχύουν τα ακόλουθα:

- Κάθε είσοδος, δεδομένη ή μη, που μπορεί να δεχθεί η συνάρτηση χαρακτηρίζεται ως **στιγμιότυπο** (instance), δημιουργώντας έτσι ένα **σύνολο στιγμιότυπων**.
- Οι εισοδοί περιγράφονται με βάση τα **γνωρίσματα** (attributes) που διαθέτουν και έχουν χαρακτηριστεί ως σημαντικά από την αρχή της μελέτης του προβλήματος που καλείται να επιλύσει το σύστημα.
- Οι δεδομένες εισοδοί συγκεντρώνονται από παρατηρήσεις και αποτελούν το λεγόμενο **σύνολο εκπαίδευσης** (training set) που αποτελεί υποσύνολο του συνόλου στιγμιότυπων.
- Το υπόλοιπο μέρος του συνόλου στιγμιότυπων αποτελεί το **σύνολο ελέγχου** (test set) που θα χρησιμοποιηθεί κατά τη φάση πιστοποίησης.
- Η συνάρτηση που απεικονίζει μια είσοδο από το σύνολο εκπαίδευσης στη γνωστή της έξοδο καλείται **συνάρτηση στόχου** (goal function).
- Η τιμή που επιστρέφει η συνάρτηση στόχου για ένα στιγμιότυπο από το σύνολο στιγμιότυπων, δίνεται σε μια μεταβλητή που καλείται **μεταβλητή στόχου** (goal variable).
- Στην επιβλεπόμενη μάθηση, η συμπεριφορά της συνάρτησης στόχου βελτιώνεται μέσω διαδικασιών εκπαίδευσης με τη βοήθεια της **συνάρτησης λάθους** (error function) που εντοπίζει τη διαφορά της μεταβλητής στόχου από την επιθυμητή έξοδο.

Εναλλακτική ορολογία που χρησιμοποιείται στο κεφάλαιο αυτό:

- Στιγμιότυπα (instances)
 - Επίσης αναφέρονται ως παραδείγματα ή δείγματα (examples)
- Γνωρίσματα (attributes)
 - Επίσης γνωστά ως χαρακτηριστικά (features), προσόντα (qualifications), μεταβλητές (variables), ανεξάρτητες μεταβλητές (independent variables), μεταβλητές εισόδου (input variables), συμεταβλητές (covariates)
- Μεταβλητή στόχου (goal variable)

- Επίσης γνωστή ως κατηγορημα στόχου (goal predicate), εξαρτημένη μεταβλητή (dependent variable), τάξη κατηγοριοποίησης (classification class)
- Συνάρτηση λάθους (error function)
 - Επίσης γνωστή ως αντικειμενική συνάρτηση (objective function), συνάρτηση απώλειας (loss function).

Αν υποθέσουμε ότι:

- h η συνάρτηση πρόγνωσης που θέλουμε να “μάθουμε”,
- f η συνάρτηση στόχου,
- D το σύνολο εκπαίδευσης,
- x ένα στιγμιότυπο σε μορφή ενός διάνυσματος q των χαρακτηριστικών του.

Τότε, καταρχάς, η έμμεση απεικόνιση του x προς το $f(x)$ μάς είναι άγνωστη. Διαθέσιμα είναι μόνο ζεύγη εκπαίδευσης του τύπου

$$D = \{(x, f(x))\}$$

Αυτό που θέλουμε να μάθουμε είναι η συνάρτηση $h(x; q)$ που υλοποιεί μια γενικευμένη απεικόνιση από το x στο f .

Αν $h(x; q)$ είναι “κοντά” στο $f(x)$ για όλα τα στιγμιότυπα x του συνόλου ελέγχου, συμπεραίνουμε ότι τα χαρακτηριστικά q είναι οι παράμετροι που πρέπει να λάβει υπόψη της η συνάρτηση πρόβλεψης $h(\cdot)$. Ένα παράδειγμα συνάρτησης πρόγνωσης δίνεται παρακάτω, όπου $q(x_1 \ x_2 \ x_3)$ το διάνυσμα των χαρακτηριστικών του x :

$$h(x; q) = (x_1 \text{ or } x_2) \text{ and } (\text{not } x_3)$$

Μια εμπειρική συνάρτηση λάθους E θα μπορούσε να υπολογίζεται ως ακολούθως:

$$E(h) = \sum_x \text{distance}[h(x; q), f]$$

όπου *distance* είναι η συνάρτηση που υπολογίζει τη διαφορά της πρόγνωσης από την πραγματική τιμή που επιστρέφει η συνάρτηση στόχου f για ένα x από το σύνολο εκπαίδευσης. Η E επιστρέφει το άθροισμα όλων των διαφορών που αφορούν τα ζεύγη εκπαίδευσης μέσα στο D .

Γνωστότεροι αλγόριθμοι Επιβλεπόμενης Επαγωγικής Μάθησης είναι:

- Δένδρα Απόφασης (DecisionTrees),
- Μάθηση βασισμένη σε Επεξηγήσεις (Explanation-Based Learning),
- Μάθηση βασισμένη σε Περιπτώσεις (Case-Based Learning),
- Μάθηση Νευρωνικών δικτύων (π.χ. για Backpropagation Neural Networks),
- Μάθηση Μέσω Στατιστικών Μεθόδων (π.χ. μάθηση κατά Bayes),
- Συλλογική Μάθηση από Ενδυνάμωση (Boosting) κ.ά.

Στη συνέχεια θα αναφερθούμε στα Δένδρα Απόφασης.

4.3.1 Δένδρα Απόφασης

Τα **Δένδρα Απόφασης-ΔΑ** (Decision Trees) είναι ο γνωστότερος αλγόριθμος επιβλεπόμενης Επαγωγικής Μάθησης και έχει εφαρμοστεί με επιτυχία σε πολλούς τομείς όπου απαιτείται ταξινόμηση: ενδεικτικά, στην αναγνώριση προσώπων σε εικόνες, στην ιατρική για διάγνωση περιστατικών, για προβλέψεις απαραίτητες

στη διαφήμιση, για προώθηση προϊόντων και, γενικότερα, για εξόρυξη γνώσης. Ο αλγόριθμος ΔΑ οδηγεί στη δημιουργία μιας δενδροειδούς μορφής που τα φύλλα της αποτελούν **κατηγορίες ταξινόμησης** (classes). Η δενδροειδής αυτή μορφή μπορεί να αναγνωστεί και ως ένα σύνολο κανόνων που καλούνται **κανόνες ταξινόμησης** (classification rules) και να δώσει μια πειστική απάντηση στο ερώτημα:

Πώς μπορεί μία μηχανή να δημιουργήσει γενικούς κανόνες από συγκεκριμένες παρατηρήσεις και πόσο αξιόπιστοι είναι αυτοί οι κανόνες στην πράξη;

Βασικές προϋποθέσεις για τη λειτουργία ενός αλγόριθμου επαγωγικής μάθησης είναι:

- καθορισμός ενός **συνόλου χαρακτηριστικών** (features set - **FS**) ως των προϋποθέσεων του επιδιωκόμενου προς εξαγωγή κανόνα ταξινόμησης:

$$FS = \{F_1, F_2, F_3, \dots, F_{|FS|}\}$$

- ύπαρξη προκαθορισμένων **διακριτών κατηγοριών ταξινόμησης** (classes - **C**) ως στόχου του διαχωρισμού τον οποίο θα επιδιώξει ο αλγόριθμος και, στη συνέχεια, ως συμπερασμάτων (conclusions) των κανόνων στους οποίους θα οδηγήσει η αναγνώριση της δενδροειδούς μορφής που θα αναπτύξει ο αλγόριθμος:

$$C = \{c_1, c_2, c_3, \dots, c_{|C|}\}$$

- ύπαρξη επαρκούς αριθμού **δειγμάτων** που θα προκύψουν από παρατηρήσεις και θα χρησιμοποιηθούν για τη δημιουργία του εκπαιδευτικού συνόλου (training set - **TS**).

Παράδειγμα προβλήματος που μπορεί να λύσει ένας αλγόριθμος ΔΑ

Έστω ότι μια διαφημιστική εταιρία έχει σκοπό να ετοιμάσει μια αφίσα, για να διαφημίσει ένα αντηλιακό, και θέλει να επιλέξει για την αφίσα ένα μοντέλο, άντρα ή γυναίκα, που θα φωτογραφηθεί σε μια παραλία εκτεθειμένο κάτω από τον ήλιο και παρόλα αυτά να μην έχει καεί .

Το μεγαλύτερο πρόβλημα που συνήθως προκύπτει και απαιτεί επίλυση είναι η δυσχέρεια στην εύρεση των εμφανισιακών χαρακτηριστικών που πρέπει να έχει το μοντέλο, για να είναι πειστικό στο ρόλο του. Αν δεν υπάρχει διαθέσιμη σχετική γνώση ή η υπάρχουσα δεν είναι ολοκληρωμένη, ώστε να δώσει απάντηση στο παραπάνω πρόβλημα, τότε απαιτείται η παραγωγή νέας γνώσης με επαγωγικό τρόπο συνήθως μέσα από παρατηρήσεις.

Στην περίπτωση κατά την οποία επιλεγεί ο αλγόριθμος των ΔΑ, τότε πρέπει να καθοριστούν:

- ως πρώτο βήμα πρέπει να αποφασιστούν οι κατηγορίες ταξινόμησης στις οποίες θα καταλήξει ο διαχωρισμός: στο πρόβλημά μας, οι κατηγορίες ταξινόμησης είναι δύο, αν το δείγμα που αφορά η παρατήρηση έχει καεί από τον ήλιο ή όχι.
- στη συνέχεια, τα υπόλοιπα εμφανισιακά χαρακτηριστικά που πρέπει να παρατηρηθούν κατά τη συγκέντρωση δειγμάτων και τα οποία θα χρησιμοποιήσει ο αλγόριθμος σε κάθε βήμα διαχωρισμού, ώστε να καταλήξει σε ένα σύνολο δειγμάτων με κοινά χαρακτηριστικά: στο παράδειγμά μας, θα είναι το χρώμα των μαλλιών, των ματιών και του δέρματος, και, επιπλέον, αν το δείγμα έχει χρησιμοποιήσει αντηλιακό ή όχι.

Βάσει των επιλεγμένων χαρακτηριστικών και των κατηγοριών ταξινόμησης, θα ακολουθήσει η συγκέντρωση των δειγμάτων που συνήθως γίνεται δια της παρατήρησης ή από συμπλήρωση ερωτηματολογίων. Στο παράδειγμά μας, η συγκέντρωση θα γίνει σε συνδυασμό, δηλαδή ο υπεύθυνος συγκέντρωσης θα παρατηρεί λουόμενους, θα σημειώνει τα χαρακτηριστικά τους, θα τους ρωτά για τη χρήση

αντηλιακού και θα τους κατατάσσει σε κατηγορίες καμένων και μη καμένων. Στον πίνακα 4.1 έχει αποτυπωθεί ένα τέτοιο μικρό σύνολο δειγμάτων εκπαίδευσης για τις ανάγκες παρουσίασης της μεθόδου.

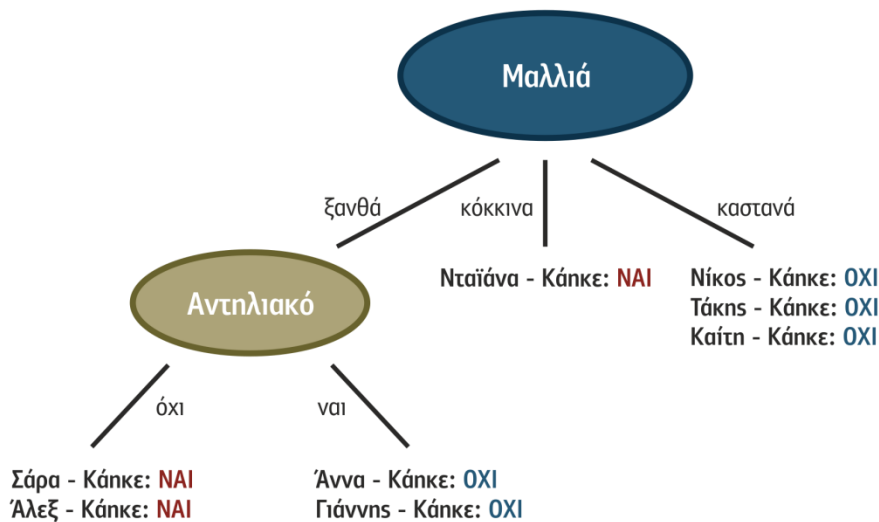
Δείγμα	Χαρακτηριστικά				Κατηγορία Ταξινόμησης
	Μαλλιά	Μάτια	Δέρμα	Αντηλιακό	Κάηκε
Σάρα	ξανθά	πράσινα	σκούρο	όχι	ΝΑΙ
Άννα	ξανθά	μπλε	ανοιχτό	ναι	ΟΧΙ
Νίκος	καστανά	μαύρα	σκούρο	ναι	ΟΧΙ
Άλεξ	ξανθά	μαύρα	ανοιχτό	όχι	ΝΑΙ
Νταϊάνα	κόκκινα	πράσινα	ανοιχτό	όχι	ΝΑΙ
Τάκης	καστανά	μπλε	σκούρο	όχι	ΟΧΙ
Καίτη	καστανά	πράσινα	ανοιχτό	όχι	ΟΧΙ
Γιάννης	ξανθά	μαύρα	ανοιχτό	ναι	ΟΧΙ

Πίνακας 4.1 Συγκεντρωτικός πίνακας δειγμάτων για εκπαίδευση αλγόριθμου ΔΑ

Σε ένα σύνολο παραμέτρων, όπου τα χαρακτηριστικά είναι πολλά και παίρνουν πολλές τιμές, οι δυνατοί συνδυασμοί που μπορεί να προκύψουν είναι πολλοί. Στο παράδειγμά μας, το χαρακτηριστικό “Μαλλιά” παίρνει 3 τιμές (ξανθά, καστανά, κόκκινα), το χαρακτηριστικό “Μάτια” 3 τιμές (πράσινα, μπλε, μαύρα), το χαρακτηριστικό “Δέρμα” 2 τιμές (σκούρο, ανοιχτό) και το χαρακτηριστικό “Αντηλιακό” 2 τιμές (ναι, όχι). Οι συνδυασμοί είναι τριάντα έξι ($3 \times 3 \times 2 \times 2 = 36$). Επομένως, αν έχει δοθεί το σύνολο των 8 δειγμάτων του ανωτέρω πίνακα προς εκπαίδευση σε ένα σύστημα που κάνει αναγνώριση χρησιμοποιώντας πιθανότητες, μία νέα περίπτωση έχει πιθανότητα να αναγνωριστεί περίπου 22%:

$$\frac{8}{36} = \sim 0.22 \rightarrow 22\%$$

Τα ΔΑ έρχονται να δώσουν λύση στην αναγνώριση της νέας περίπτωσης, παράγοντας γενικευμένους κανόνες αναγνώρισης που στην περίπτωσή μας θα καθορίσουν το πότε κάποιος καίγεται από τον ήλιο και πότε όχι και θα αποτελούσαν μία λύση στην προσπάθεια της μηχανής να κατατάσσει κάθε νέα περίπτωση χρησιμοποιώντας αυτόν τον κανόνα. Ένα τέτοιο επιτυχημένο δένδρο αναγνώρισης για το πρόβλημά μας είναι αυτό του σχήματος 4.2.



Σχήμα 4.2 Δένδρο απόφασης για το πρόβλημα της διαφήμισης αντηλιακού

Στην ουσία, ο αλγόριθμος των ΔΑ δημιουργεί μια δενδροειδή μορφή όπως προαναφέραμε, σε κάθε κόμβο της οποίας επιχειρείται ο περαιτέρω διαχωρισμός των μη κατηγοριοποιημένων δειγμάτων του συνόλου εκπαίδευσης που έχουν συγκεντρωθεί εκεί. Η δενδροειδής αυτή μορφή καλείται **δένδρο απόφασης** (decision tree) και όταν ο αλγόριθμος πετυχαίνει έναν πλήρη διαχωρισμό δειγμάτων τότε στα φύλλα της συγκεντρώνονται μόνο ομοιογενή δείγματα, δηλαδή δείγματα της ίδιας κατηγορίας. Κάθε φύλλο του ΔΑ με ομοιογενή δείγματα επιτρέπει την **εξαγωγή ενός συμπεράσματος** που μπορεί να εκφραστεί ως κανόνας που εκφράζει τον τρόπο προσδιορισμού μιας συγκεκριμένης κατηγορίας βάσει των χαρακτηριστικών της. Για παράδειγμα, το αριστερότερο φύλλο του δένδρου στο σχήματος 2.2 περιέχει ομοιογενή δείγματα άρα το συμπέρασμα που εξάγεται υπό μορφή κανόνα είναι το ακόλουθο:

ΑΝ Μαλλιά ξανθά
και Αντηλιακό όχι
ΤΟΤΕ περίπτωση καίγεται

Εάν για το πρόβλημα του παραδείγματός μας υπάρχουν 10 δείγματα συγκεντρωμένα μέσα στο σύνολο εκπαίδευσης, όπως αυτά που είναι καταχωρημένα μέσα στον πίνακα 4.2, τότε ο αλγόριθμος θα δημιουργήσει ένα δένδρο απόφασης σαν αυτό που παρουσιάζεται στο σχήμα 4.2, όπου δίπλα σε κάθε εσωτερικό κόμβο αναφέρονται τα δείγματα που αναμένουν για διαχωρισμό και δίπλα σε κάθε φύλλο αναφέρονται τα πλήρως διαχωρισμένα δείγματα, δηλαδή δείγματα μιας ζητούμενης κατηγορίας (ΝΑΙ, ΟΧΙ).

Για την περιγραφή ενός δένδρου αναγνώρισης μπορούμε να πούμε ότι:

- Η ρίζα του δένδρου είναι ένα από τα χαρακτηριστικά που ο αλγόριθμος κρίνει προσφορότερο να επιλέξει πρώτα.
- Κάθε εσωτερικός κόμβος του δένδρου ονοματίζεται με το όνομα ενός νέου χαρακτηριστικού που δεν έχει ήδη χρησιμοποιηθεί στο συγκεκριμένο κλαδί του δένδρου.
- Κάθε ακμή ονοματίζεται με μια διαφορετική τιμή που μπορεί να πάρει το χαρακτηριστικό του κόμβου από τον οποίο ξεκινάει.
- Κάθε φύλλο αντιστοιχεί σε μια κατηγορία ταξινόμησης.

Για τη δημιουργία ενός δένδρου απόφασης, ο αλγόριθμος χρησιμοποιεί την τεχνική “διαίρει και βασίλευε”, όπως φαίνεται και στον αλγόριθμο που ακολουθεί:

Αλγόριθμος δένδρου απόφασης

Έστω **TS** το σύνολο των δειγμάτων εκπαίδευσης

Αν

όλα τα δείγματα του **TS** ανήκουν στην ίδια κατηγορία **C**,

Τότε

το δέντρο απόφασης του **TS** είναι φύλλο που αντιστοιχεί στην κατηγορία **c**.

Αλλιώς

Αν

τα δείγματα στο **TS** ανήκουν σε διαφορετικές κατηγορίες και **FS** το σύνολο των χαρακτηριστικών του **TS**, και **F_i** ένα επιλεγμένο χαρακτηριστικό του οποίου οι τιμές **v₁, v₂, v₃, ... v_n** διαμερίζουν το σύνολο **TS** σε υποσύνολα **TS₁, TS₂, TS₃, ... TS_n**

Τότε

το δέντρο απόφασης για το **TS** είναι ο κόμβος που εξετάζει το **F_i** και κάτω από τον κόμβο αυτό θα κατασκευαστούν **n** υποδέντρα, ένα για κάθε τιμή **v₁, v₂, v₃, ... v_n**.

Σε κάθε υποδέντρο κατανέμονται τα υποσύνολα **TS₁, TS₂, TS₃, ... TS_n**

Για κάθε **TS_j** υποσύνολο

Αν

όλα τα δείγματα ανήκουν στην ίδια κατηγορία,

Τότε

κατασκευάζεται ένα φύλλο για το **TS_j** που αντιστοιχεί στην κατηγορία αυτή.

Αλλιώς,

κατασκευάζεται ένας κόμβος για το **TS_j** που εξετάζει ένα άλλο χαρακτηριστικό από το σύνολο **FS' = FS - {F_i}**

Αν

FS' είναι κενό,

Τότε

κατασκευάζεται φύλλο για το **TS_j** που αντιστοιχεί στην πιο κοινή κατηγορία **c** των δειγμάτων του **TS_j**

όπου **C** κατηγορία ταξινόμησης

TS το σύνολο δειγμάτων εκπαίδευσης

FS το σύνολο των χαρακτηριστικών του **TS**, και

F_i ένα επιλεγμένο χαρακτηριστικό του οποίου οι τιμές **v₁, v₂, v₃, ... v_n** διαμερίζουν το σύνολο **TS** σε υποσύνολα **TS₁, TS₂, TS₃, ... TS_n**

Η σωστή επιλογή χαρακτηριστικών, ο επαρκής αριθμός αυτών και των συγκεντρωμένων δειγμάτων αποτελούν κλειδί για μια αποτελεσματική λειτουργία του αλγόριθμου.

Ένας από τους πιο γνωστούς αλγόριθμους δημιουργίας Δένδρων Απόφασης είναι ο ID3 (Quinlan, 1986).

Ο αλγόριθμος ID3

Ο αλγόριθμος ID3 (Iterative Dichotomizer 3) αναπτύχθηκε αρχικά στο Πανεπιστήμιο του Σίδνεϋ από τον J. Ross Quinlan και βασίζεται στον αλγόριθμο Concept Learning System (CLS) που αποτελεί τυπικό αλγόριθμο δημιουργίας ΔΑ.

Αυτό που διαθέτει επιπλέον ο ID3 είναι η ευρετική αναζήτηση χαρακτηριστικού για διαχωρισμό. Δηλαδή, σε κάθε κόμβο του δένδρου αναζητά μεταξύ των χαρακτηριστικών του συνόλου δειγμάτων

εκπαίδευσης αυτό το χαρακτηριστικό το οποίο διαχωρίζει καλύτερα τα δεδομένα δείγματα. Εάν το χαρακτηριστικό διαχωρίζει πλήρως το σύνολο εκπαίδευσης, τότε ο ID3 σταματά. Αλλιώς, λειτουργεί αναδρομικά στα n (όπου n = αριθμός των πιθανών τιμών ενός χαρακτηριστικού) διαχωρισμένα υποσύνολα, για να εντοπίσει το “καλύτερο” τους χαρακτηριστικό. Ο αλγόριθμος εφαρμόζει αναδρομικά μια *άπληστη αναζήτηση*, δηλαδή επιλέγει το καλύτερο χαρακτηριστικό και δεν ανατρέχει σε προηγούμενα χαρακτηριστικά που έχει χρησιμοποιήσει στη δεδομένη πορεία του για να τα επανεξετάσει.

Τα χαρακτηριστικά του αλγόριθμου ID3 είναι τα ακόλουθα:

- Μεγαλώνει επαναληπτικά ένα μικρό σύνολο εκπαίδευσης από δεδομένα που υπάρχουν σε μία περιορισμένη βάση δειγμάτων.
- Αφαιρεί χαρακτηριστικά που περιέχονται στην κατηγορία προς ταξινόμηση με σκοπό τη συμπίεση της βάσης.
- Χρησιμοποιεί στατιστικές μεθόδους (Shannon's Information Statistic, *μεγέθη εντροπία και κέρδος πληροφορίας*), για να επιλέξει χαρακτηριστικά για διαχωρισμό.

Ο ID3, όπως και κάθε άλλος επαγωγικός αλγόριθμος, μπορεί να κατηγοριοποιήσει λανθασμένα ένα νέο στιγμιότυπο για πολλούς διαφορετικούς λόγους. Ο πιο απλός είναι να μην έχει γίνει καλός σχεδιασμός και η συγκέντρωση των δειγμάτων να μην έχει βασιστεί σε σωστά χαρακτηριστικά. Το πρόβλημα δεν αφορά τον αλγόριθμο, αλλά οδηγεί σε κακής ποιότητας συμπεράσματα στα φύλλα του δένδρου που παράγει. Άλλος λόγος είναι ότι μη επαρκές πλήθος των δειγμάτων θα οδηγήσει σε πρόωρες συγκλίσεις και υπεραπλουστευμένα συμπεράσματα. Τέλος, βασικότερο πρόβλημα για το οποίο ευθύνεται ο αλγόριθμος είναι η κακή επιλογή μεθόδου για τον ευρετικό ή συνήθως στατιστικό εντοπισμό του “καλού” χαρακτηριστικού.

Επιλογή χαρακτηριστικού

Προκειμένου να αποφασιστεί η καλύτερη επιλογή χαρακτηριστικού, ο αλγόριθμος ID3 βασίζεται στις έννοιες *εντροπία και κέρδος πληροφορίας*.

Εντροπία

Η *εντροπία πληροφορίας* (information entropy) χαρακτηρίζει το βαθμό αβεβαιότητας ενός συνόλου δεδομένων S : ενός συνόλου δεδομένων S :

$$Entropy(S) = \sum_{i=1}^n -p_i \log p_i$$

όπου p_1, p_2, \dots, p_i οι πιθανότητες του κάθε ενδεχομένου που περιλαμβάνεται στο σύνολο.

Ο αλγόριθμος ID3 χρησιμοποιεί την εντροπία ως μέτρο ομοιογένειας των τιμών του αρχικού συνόλου δεδομένων προς κάποιο χαρακτηριστικό, καθώς και τα υποσύνολα που προκύπτουν σε κάθε βήμα μετά την ταξινόμηση.

Εάν όλα τα δείγματα του συνόλου εκπαίδευσης είναι ομοιογενή ως προς μια κατηγορία, τότε η εντροπία του ισούται με μηδέν. Εάν οι κατηγορίες είναι διαφορετικές και τα δείγματα που ανήκουν σε καθεμία έχουν το ίδιο πλήθος, τότε η εντροπία είναι 1.

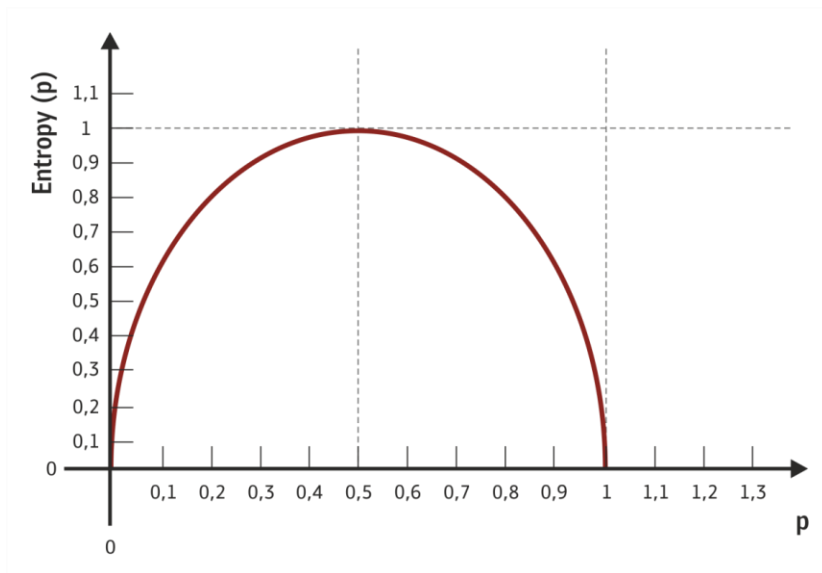
Στη δυαδική περίπτωση, όπου οι κατηγορίες είναι δύο, η εντροπία δίνεται από τη συνάρτηση:

$$Entropy(p) = -p \log p - (1-p) \log (1-p)$$

Παράδειγμα, έστω δοχείο με N μπάλες με πιθανότητα ενδεχομένου η μπάλα να είναι λευκή $N(p)$ και να είναι μαύρη $N(1-p)$. Τότε ισχύει:

- Αν οι μπάλες είναι όλες μαύρες ή όλες άσπρες, δηλαδή $p=1$ ή $(1-p)=1$, τότε Εντροπία=0

- Αν οι μπάλες είναι μισές μαύρες και μισές άσπρες, δηλαδή $p=(1-p)=\frac{1}{2}$ (πιθανότητα επιλογής 50%) τότε Εντροπία=1 (μέγιστη)



Σχήμα 4.3 Εντροπία σε μια δυαδική περίπτωση κατηγοριών

Εάν σε ένα σύνολο S 14 μπαλών υπήρχαν 5 άσπρες μπάλες και 9 μαύρες, τότε η εντροπία θα υπολογιζόταν ως εξής:

$$\begin{aligned} Entropy(S) &= \sum_{i=1}^2 -p_i \log p_i = -\frac{5}{14} \log \frac{5}{14} - \frac{9}{14} \log \frac{9}{14} \\ &= -0,36 \log 0,36 - 0,64 \log 0,64 = 0,94 \end{aligned}$$

Μπάλες	
Μαύρες	Άσπρες
9	5

$$\begin{aligned} Entropy(\text{Μπάλες}) &= Entropy(5,9) \\ &= Entropy(0.36,0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

Σχήμα 4.4 Εύρεση εντροπίας με την χρήση πίνακα συχνοτήτων ενός χαρακτηριστικού

Έστω τώρα ότι μας ενδιαφέρει η εντροπία ενός χαρακτηριστικού X με n διαφορετικές τιμές c σε σχέση με την κατηγορία S. Τότε η εντροπία δίνεται με τη συνάρτηση:

$$Entropy(S, X) = \sum_{c=1}^n P(c) * Entropy(c)$$

Στο σχήμα 4.5 δίνεται ένα παράδειγμα υπολογισμού της εντροπίας για το πρόβλημα της διαφήμισης αντηλιακού που παρουσιάστηκε παραπάνω, με βάση το σύνολο εκπαίδευσης που δόθηκε στον [πίνακα 4.1](#):

		Κάπκε		
		ΝΑΙ	ΟΧΙ	
Μαλλιά	μαύρα	3	2	5
	κόκκινα	4	0	4
	ξανθά	2	3	5
				14



$$\begin{aligned}
 \text{Entropy (Κάπκε, Μαλλιά)} &= P(\text{μαύρα}) \\
 &\quad * \text{Entropy}(3,2) \\
 &+ P(\text{κόκκινα}) \\
 &\quad * \text{Entropy}(4,0) \\
 &+ P(\text{ξανθά}) \\
 &\quad * \text{Entropy}(2,3) \\
 &= \frac{5}{14} * 0,971 + *0,0 \\
 &+ \frac{5}{14} * 0,971 = 0,693
 \end{aligned}$$

Σχήμα 4.5 Εύρεση εντροπίας με τη χρήση πίνακα συχνοτήτων δύο χαρακτηριστικών

$$\begin{aligned}
 \text{Entropy(Κάπκε, Μαλλιά)} \\
 &= P(\text{μαύρα}) * \text{Entropy}(3,2) + P(\text{κόκκινα}) * \text{Entropy}(4,0) + P(\text{ξανθά}) \\
 &\quad * \text{Entropy}(2,3) = \frac{5}{14} * 0.971 + \frac{4}{14} * 0.0 + \frac{5}{14} * 0.971 = 0.693
 \end{aligned}$$

Κέρδος Πληροφορίας

Εάν A είναι χαρακτηριστικό σε ένα S σύνολο δειγμάτων από το σύνολο μάθησης, τότε το **κέρδος πληροφορίας** (information gain) χαρακτηρίζει το πόση πληροφορία “φέρει” το χαρακτηριστικό A :

$$G(S, A) = E(S) - \sum_{i=1}^m f_s(A_i) * E(S_{A_i})$$

όπου

- $E(\dots)$ η συνάρτηση εντροπίας
- m το πλήθος των τιμών A_i που παίρνει το A στο S
- $f_s(A_i)$ το ποσοστό των δειγμάτων στο S που παίρνουν την τιμή A_i
- S_{A_i} το υποσύνολο του S όπου η τιμή του A είναι A_i

Ή πιο απλά:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \text{Entropy}(S, A)$$

Η γενική ιδέα του ID3, για να διαχωρίσει ένα μη ομοιογενές σύνολο δειγμάτων σε έναν κόμβο του δένδρου απόφασης που ανήκει σε ένα συγκεκριμένο κλαδί, είναι:

- Για όλα τα αχρησιμοποίητα στο συγκεκριμένο κλαδί χαρακτηριστικά υπολόγισε την εντροπία σε σχέση με τα δείγματα.
- Διάλεξε το χαρακτηριστικό που παρουσιάζει το μέγιστο κέρδος πληροφορίας.
- Κατασκεύασε κόμβο για το χαρακτηριστικό αυτό.

Παράδειγμα ID3

Αν χρησιμοποιήσουμε τον αλγόριθμο ID3 για το πρόβλημα της διαφήμισης αντηλιακού που παρουσιάσαμε προηγουμένως, τα χαρακτηριστικά (FS) που θα χρησιμοποιήσει ο αλγόριθμος θα είναι:

- Μαλλιά = {ξανθά, καστανά, κόκκινα}
- Μάτια = {καφέ, μπλε, πράσινα}
- Δέρμα = {ανοιχτό, σκούρο}
- Αντηλιακό = {ναι, όχι}

Οι κατηγορίες ταξινόμησης (C) είναι:

- Κήκε = {ΝΑΙ, ΟΧΙ}

Το σύνολο δειγμάτων εκπαίδευσης είναι συγκεντρωμένο στον επόμενο πίνακα 4.2.

Δείγμα	Χαρακτηριστικά				Κατηγορία Ταξινόμησης
	Μαλλιά	Μάτια	Δέρμα	Αντηλιακό	Κήκε
T1	ξανθά	καφέ	σκούρο	όχι	ΟΧΙ
T2	ξανθά	καφέ	σκούρο	ναι	ΟΧΙ
T3	κόκκινα	καφέ	σκούρο	όχι	ΝΑΙ
T4	μαύρα	μπλε	σκούρο	όχι	ΝΑΙ
T5	μαύρα	πράσινα	ανοιχτό	όχι	ΝΑΙ
T6	μαύρα	πράσινα	ανοιχτό	ναι	ΟΧΙ
T7	κόκκινα	πράσινα	ανοιχτό	ναι	ΝΑΙ
T8	ξανθά	μπλε	σκούρο	όχι	ΟΧΙ
T9	ξανθά	πράσινα	ανοιχτό	όχι	ΝΑΙ
T10	μαύρα	μπλε	ανοιχτό	όχι	ΝΑΙ
T11	ξανθά	μπλε	ανοιχτό	ναι	ΝΑΙ
T12	κόκκινα	μπλε	σκούρο	ναι	ΝΑΙ
T13	κόκκινα	καφέ	ανοιχτό	όχι	ΝΑΙ
T14	μαύρα	μπλε	σκούρο	ναι	ΟΧΙ

Πίνακας 4.2 Σύνολο εκπαίδευσης για το πρόβλημα της διαφήμισης αντηλιακού

Παράδειγμα δημιουργίας δένδρου απόφασης με χρήση εντροπίας και κέρδους πληροφορίας

Βήμα 1: Υπολογισμός εντροπίας της κατηγορίας του στόχου

$$\begin{aligned}
 Entropy(Κάηκε) &= Entropy(5,9) \\
 &= Entropy(0.36, 0.64) \\
 &= -(0.36 \log 0.36) - (0.64 \log 0.64) \\
 &= 0.94
 \end{aligned}$$

Βήμα 2: Το σύνολο εκπαίδευσης διαχωρίζεται ως προς τα διαφορετικά χαρακτηριστικά.

Για κάθε διαφορετικό χαρακτηριστικό γίνεται ο ακόλουθος υπολογισμός: Υπολογίζεται η εντροπία για κάθε διαφορετική τιμή του χαρακτηριστικού και στη συνέχεια, αφού πολλαπλασιαστεί με το ποσοστό των δειγμάτων που διαθέτουν την αντίστοιχη τιμή, προστίθεται για να υπολογιστεί η συνολική εντροπία του διαχωρισμένου τμήματος.

$$\begin{aligned}
 Entropy(Κάηκε, Μαλλιά) &= P(\text{μαύρα}) * Entropy(3,2) \\
 &+ P(\text{κόκκινα}) * Entropy(4,0) \\
 &+ P(\text{ξανθά}) * Entropy(2,3) \\
 &= \frac{5}{14} * 0.971 + \frac{4}{14} * 0.0 + \frac{5}{14} * 0.971 \\
 &= 0.693
 \end{aligned}$$

Η εντροπία που προκύπτει ως αποτέλεσμα αφαιρείται από την εντροπία πριν από το διαχωρισμό. Το αποτέλεσμα είναι το κέρδος πληροφορίας (Gain).

$$\begin{aligned}
 Gain(Κάηκε, Μαλλιά) &= Entropy(Κάηκε) - Entropy(Κάηκε, Μαλλιά) \\
 &= 0.940 - 0.693 = 0.247
 \end{aligned}$$

Το κέρδος πληροφορίας πρέπει να υπολογιστεί για κάθε διαφορετικό χαρακτηριστικό που μπορεί να εφαρμοστεί για να διαχωρίσει ένα συγκεκριμένο υποσύνολο μη πλήρως διαχωρισμένων δειγμάτων που είναι συγκεντρωμένα σε έναν κόμβο του δένδρου (βλέπε Σχήμα 4.6).

		Κάηκε	
		ΝΑΙ	ΟΧΙ
Μαλλιά	μαύρα	3	2
	κόκκινα	4	0
	ξανθά	2	3
		Gain=0.247	

		Κάηκε	
		ΝΑΙ	ΟΧΙ
Μάτια	καφέ	2	2
	μπλε	4	2
	πράσινα	3	1
		Gain=0.029	

		Κάηκε	
		ΝΑΙ	ΟΧΙ
Δέρμα	σκούρο	3	4
	ανοιχτό	6	1
		Gain=0.152	

		Κάηκε	
		ΝΑΙ	ΟΧΙ
Αντηλιακό	όχι	6	2
	ναι	3	3
		Gain=0.048	

Σχήμα 4.6 Υπολογισμός κέρδους πληροφορίας για επιλογή χαρακτηριστικού

Βήμα 3: Επιλογή του χαρακτηριστικού με το μεγαλύτερο κέρδος.

Στον πρώτο κύκλο διαχωρισμού επιλέγεται το χαρακτηριστικό *Μαλλιά* για να διαχωρίσει το δείγμα, δεδομένου ότι έχει το μεγαλύτερο κέρδος πληροφορίας.

Παρατηρούμε ότι τα δείγματα που έχουν κόκκινα μαλλιά είναι ομοιογενή ως προς το αποτέλεσμα, δηλαδή έχουν όλα καεί.

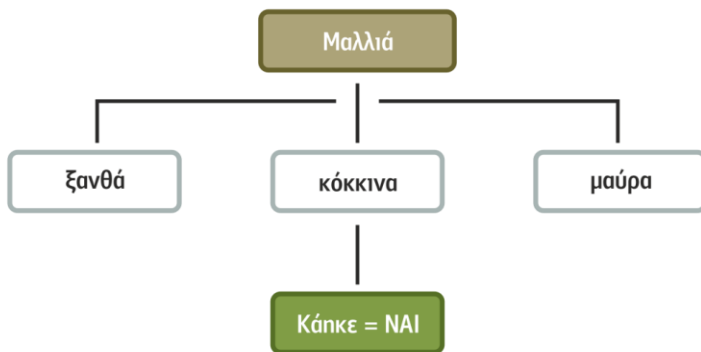
		Κάπκε	
		ΝΑΙ	ΟΧΙ
Μαλλιά	μαύρα	3	2
	κόκκινα	4	0
	ξανθά	2	3
		Gain=0.247	



Σχήμα 4.7 Επιλογή του χαρακτηριστικού *Μαλλιά* για να διαχωρίσει το δείγμα

Βήμα 4Α: Ολοκλήρωση διαχωρισμού

Ένα κλαδί με εντοπία 0, δηλαδή όταν όλα τα δείγματα του κόμβου είναι ομοιογενή, θεωρείται φύλλο. Ένα κλαδί που ξεκινά από τη ρίζα του δένδρου και καταλήγει σε φύλλο, θεωρείται ολοκληρωμένο και μπορεί να εκφραστεί ως κανόνας.



Σχήμα 4.8 Ημιτελές δένδρο απόφασης με ένα πλήρως διαχωρισμένο κλαδί.

Βήμα 4Β: Συνέχιση διαχωρισμού

Ένα κλαδί με εντροπία μεγαλύτερη από το 0, δηλαδή όταν τα δείγματα του κόμβου δεν είναι ομοιογενή, χρειάζεται περαιτέρω διαχωρισμό. Το ημιτελές δένδρο απόφασης στο σχήμα 4.8 περιέχει ένα ολοκληρωμένο κλαδί και δύο που χρήζουν περαιτέρω διαχωρισμού.

Για να διαχωρίσουμε τα ανομοιογενή δείγματα του τελευταίου κόμβου του αριστερού κλαδιού του δένδρου και επειδή χρησιμοποιήσαμε το χαρακτηριστικό *Μαλλιά* ήδη, πρέπει να αποφασίσουμε μόνο για το

ποιο από τα υπόλοιπα τρία χαρακτηριστικά είναι καταλληλότερο να εφαρμοστεί: *Μάτια*, *Δέρμα* ή *Αντηλιακό*. Μελετώντας ξεχωριστά το κέρδος πληροφορίας για κάθε ένα από αυτά, έχουμε τα παρακάτω αποτελέσματα:

Μαλλιά ξανθά = {T1, T2, T8, T9, T11} = 5 δείγματα από τον πίνακα 1 με *Μαλλιά* = *Ξανθά*

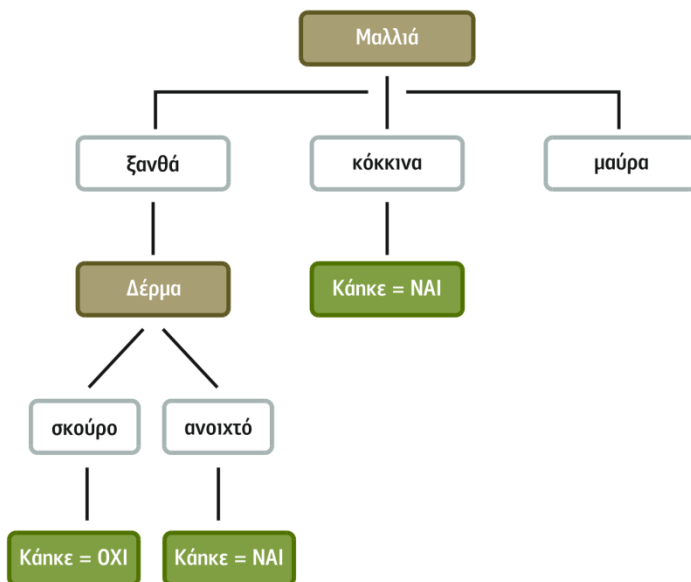
Κέρδος πληροφορίας (Μαλλιά ξανθά, Δέρμα) = 0.970

Κέρδος πληροφορίας (Μαλλιά ξανθά, Μάτια) = 0.570

Κέρδος πληροφορίας (Μαλλιά ξανθά, Αντηλιακό) = 0.019

Παρατηρούμε ότι το χαρακτηριστικό *Δέρμα* έχει το υψηλότερο κέρδος πληροφορίας. Ως εκ τούτου, εφαρμόζεται ως χαρακτηριστικό διαχωρισμού των δειγμάτων του συγκεκριμένου κόμβου (Μαλλιά Ξανθά), όπως φαίνεται στο παρακάτω σχήμα 4.9.

Δείγμα	Μαλλιά	Μάτια	Δέρμα	Αντηλιακό	Κάπκε
T1	ξανθά	καφέ	σκούρο	όχι	OXI
T2	ξανθά	καφέ	σκούρο	ναι	OXI
T8	ξανθά	μπλε	σκούρο	όχι	OXI
T9	ξανθά	πράσινα	ανοιχτό	όχι	NAI
T11	ξανθά	μπλε	ανοιχτό	ναι	NAI

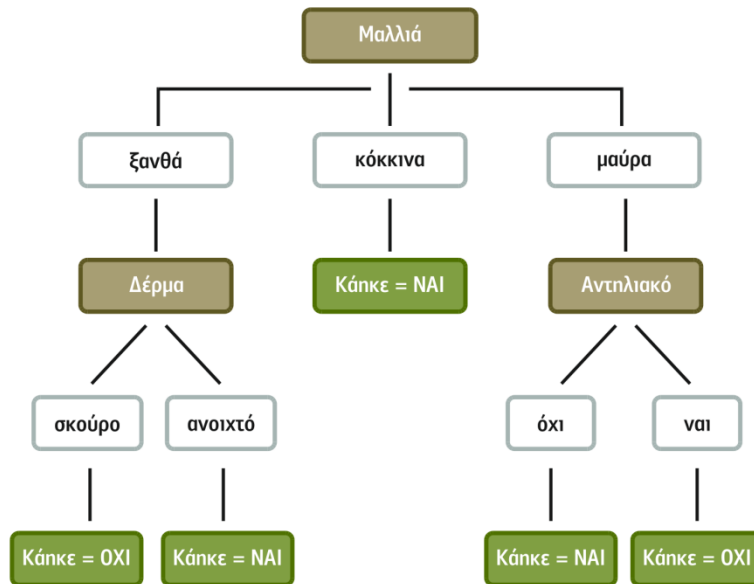


Σχήμα 4.9. Ημιτελές δένδρο απόφασης με δύο πλήρως διαχωρισμένα κλαδιά.

Βήμα 5: Ολοκλήρωση διαχωρισμού

Ο αλγόριθμος ID3 τρέχει αναδρομικά διαχωρίζοντας σε κάθε κύκλο έναν κόμβο του δένδρου που δεν είναι φύλλο, έως ότου όλα τα δείγματα κατηγοριοποιηθούν.

Δείγμα	Μαλλιά	Μάτια	Δέρμα	Αντηλιακό	Κάπκε
T4	μαύρα	μπλε	σκούρο	όχι	ΝΑΙ
T5	μαύρα	πράσινα	ανοιχτό	όχι	ΝΑΙ
T6	μαύρα	πράσινα	ανοιχτό	ναι	ΟΧΙ
T10	μαύρα	μπλε	ανοιχτό	όχι	ΝΑΙ
T14	μαύρα	μπλε	σκούρο	ναι	ΟΧΙ



Σχήμα 4.10 Δένδρο απόφασης με όλα τα δείγματα κατηγοριοποιημένα

Δείτε [κινούμενη εικόνα 4.1 - Βηματική ανάπτυξη Δένδρου Απόφασης](#)

Αν δεν υπάρχουν άλλα χαρακτηριστικά για να διαχωρίσουν έναν κόμβο που δεν είναι φύλλο, αυτό σημαίνει αποτυχία διαχωρισμού ελλείψει επαρκών χαρακτηριστικών και η όλη μελέτη για τη λύση του προβλήματος πρέπει να επαναληφθεί από την αρχή.

Βήμα 6: Έλεγχος αποτελεσμάτων

Ο αλγόριθμος μπαίνει σε φάση πιστοποίησης (recall) ελέγχοντας την εγκυρότητα των αποτελεσμάτων του με τη βοήθεια των δειγμάτων του συνόλου ελέγχου, δηλαδή δειγμάτων που δεν είχαν περιληφθεί στο σύνολο εκπαίδευσης.

Σε περίπτωση που υπάρχουν εξαιρέσεις, τα εξαιρούμενα δείγματα προστίθενται στο σύνολο εκπαίδευσης και η εκπαίδευση επαναλαμβάνεται από την αρχή.

Αν δεν υπάρχουν εξαιρέσεις, τα αποτελέσματα θεωρούνται έγκυρα και δίνονται προς χρήση.

Άλλα παραδείγματα εφαρμογής ID3

Η εντροπία και το κέρδος πληροφορίας δεν είναι ο μόνος τρόπος επιλογής χαρακτηριστικών. Σε πολλά προβλήματα ως ευρετικοί μηχανισμοί χρησιμοποιούνται άλλες απλούστερες μέθοδοι, όπως για παράδειγμα ο δοκιμαστικός διαχωρισμός των δειγμάτων του συνόλου εκπαίδευσης σε υποσύνολα βάσει κάθε διαφορετικού χαρακτηριστικού και επιλογή εκείνου που (ιδανικά) θα αποδώσει υποσύνολα με δείγματα “όλα θετικά” ή “όλα αρνητικά” ή θα προσεγγίσει μια τέτοια κατάσταση.

Παράδειγμα όπου εφαρμόζεται μια τέτοια τακτική είναι το πρόβλημα της αναμονής σε εστιατόριο (Russel & Norvig, 2003). Το συγκεκριμένο πρόβλημα αφορά τη λήψη απόφασης ενός πελάτη να περιμένει ή όχι για να φάει σε ένα εστιατόριο.

Το σύνολο των χαρακτηριστικών (FS) περιλαμβάνει τα ακόλουθα χαρακτηριστικά:

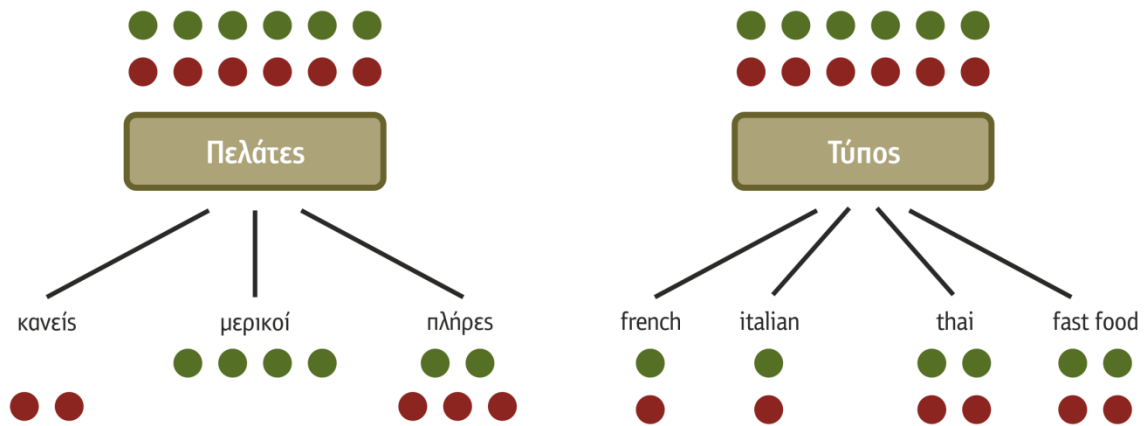
Εναλλακτικό:	υπάρχει μια εναλλακτική λύση εστιατορίου κοντά (Ναι/Όχι);
Μπαρ:	υπάρχει κοντά ένα άνετο μπαρ για να περιμένει κανείς εκεί(Ναι/Όχι)
Π/Σ:	η ημέρα είναι Παρασκευή ή Σάββατο(Ναι/Όχι)
Πεινασμένος:	ο πελάτης είναι πεινασμένος (Ναι/Όχι)
Πελάτες:	ποιο το πλήθος των πελατών στο εστιατόριο (Κανείς, Μερικοί, Πλήρες)
Τιμή:	εύρος τιμών (\$, \$\$, \$\$\$)
Βρέχει:	έξω βρέχει (Ναι/Όχι)
Κράτηση:	έχει γίνει κράτηση (Ναι/Όχι)
Τύπος:	είδος του εστιατορίου (Γαλλικό-fre, Ιταλικό-ital, Ταϊλανδέζικο-thai, Ταχυφαγείο-fast)
Αναμονή:	εκτιμώμενος χρόνος αναμονής σε λεπτά (0-10, 10-30, 30-60,> 60)

Το σύνολο εκπαίδευσης είναι καταχωρημένο στον ακόλουθο πίνακα 4.3.

Δείγμα	Χαρακτηριστικά										Αναμονή?
	Εναλλακτικό	Μπαρ	Π/Σ	Πεινασμένος	Πελάτες	Τιμή	Βρέχει	Κράτηση	Τύπος	Εκτίμηση Αναμονής	
X ₁	ναι	όχι	όχι	ναι	μερικοί	\$\$\$	όχι	ναι	fre	0-10	ΝΑΙ
X ₂	ναι	όχι	όχι	ναι	πλήρες	\$	όχι	όχι	thai	30-60	ΟΧΙ
X ₃	όχι	ναι	όχι	όχι	μερικοί	\$	όχι	όχι	fast	0-10	ΝΑΙ
X ₄	ναι	όχι	ναι	ναι	πλήρες	\$	όχι	όχι	thai	10-30	ΝΑΙ
X ₅	ναι	όχι	ναι	όχι	πλήρες	\$\$\$	όχι	ναι	fre	> 60	ΟΧΙ
X ₆	όχι	ναι	όχι	ναι	μερικοί	\$\$	ναι	ναι	ital	0-10	ΝΑΙ
X ₇	όχι	ναι	όχι	όχι	κανείς	\$	ναι	όχι	fast	0-10	ΟΧΙ
X ₈	όχι	όχι	όχι	ναι	μερικοί	\$\$	ναι	ναι	thai	0-10	ΝΑΙ
X ₉	όχι	ναι	ναι	όχι	πλήρες	\$	ναι	όχι	fast	> 60	ΟΧΙ
X ₁₀	ναι	ναι	ναι	ναι	πλήρες	\$\$\$	όχι	ναι	ital	10-30	ΟΧΙ
X ₁₁	όχι	όχι	όχι	όχι	κανείς	\$	όχι	όχι	thai	0-10	ΟΧΙ
X ₁₂	ναι	ναι	ναι	ναι	πλήρες	\$	όχι	όχι	fast	30-60	ΝΑΙ

Πίνακας 4.3 Σύνολο εκπαίδευσης για το πρόβλημα της αναμονής στο εστιατόριο

Για την επιλογή του 1^{ου} χαρακτηριστικού που θα διαχωρίσει το αρχικό σύνολο δειγμάτων του προβλήματος, έστω ότι συγκρίνονται τα χαρακτηριστικά *Πελάτες* και *Τύπος*. Από το αποτέλεσμα παρατηρούμε ότι το *Πελάτες* διαχωρίζει το σύνολο δημιουργώντας δυο φύλλα και ένα μη διαχωρισμένο κόμβο, ενώ το *Ναι/Όχι* δε δημιουργεί κανένα φύλλο (βλέπε Σχήμα 4.11).

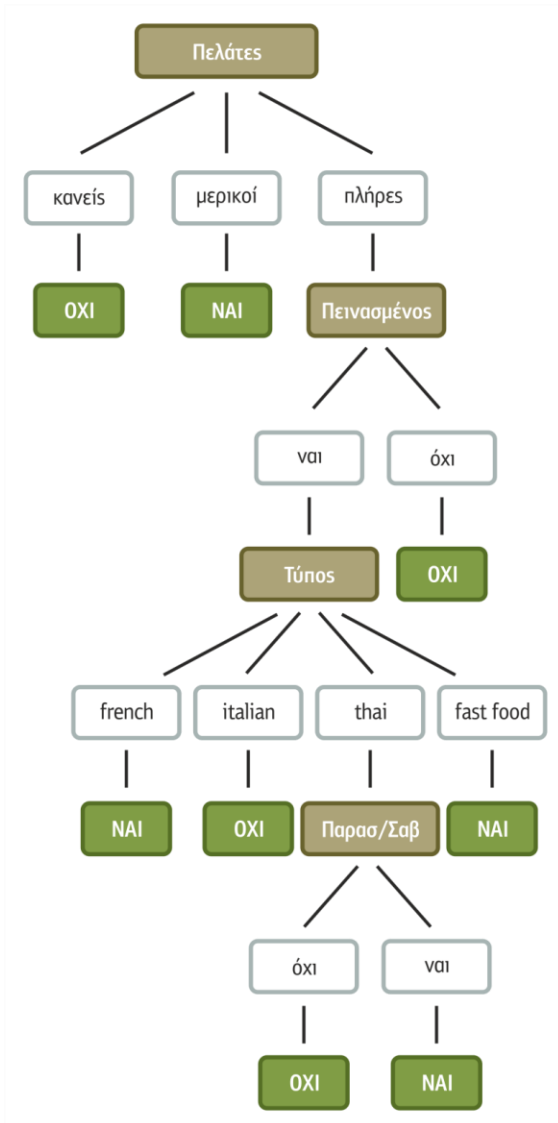


Σχήμα 4.11 Σύγκριση χαρακτηριστικών για το πρόβλημα αναμονής σε εστιατόριο

Αν ως ευρετικό κριτήριο θεωρεί ότι το καλύτερο χαρακτηριστικό είναι αυτό με το οποίο ο διαχωρισμός οδηγεί στα περισσότερα φύλλα, τότε ο αλγόριθμος θα επιλέξει το *Πελάτες* έναντι του *Τύπος*. Προφανώς, η σύγκριση πρέπει να γίνει μεταξύ όλων των χαρακτηριστικών που μπορούν να εφαρμοστούν.

Έστω ότι το χαρακτηριστικό *πελάτες* είναι καλύτερο όλων και εφαρμόζεται στην κορυφή του δένδρου και στη συνέχεια το παραπάνω ευρετικό κριτήριο εφαρμόζεται σε κάθε κόμβο που πρέπει να διαχωριστεί.

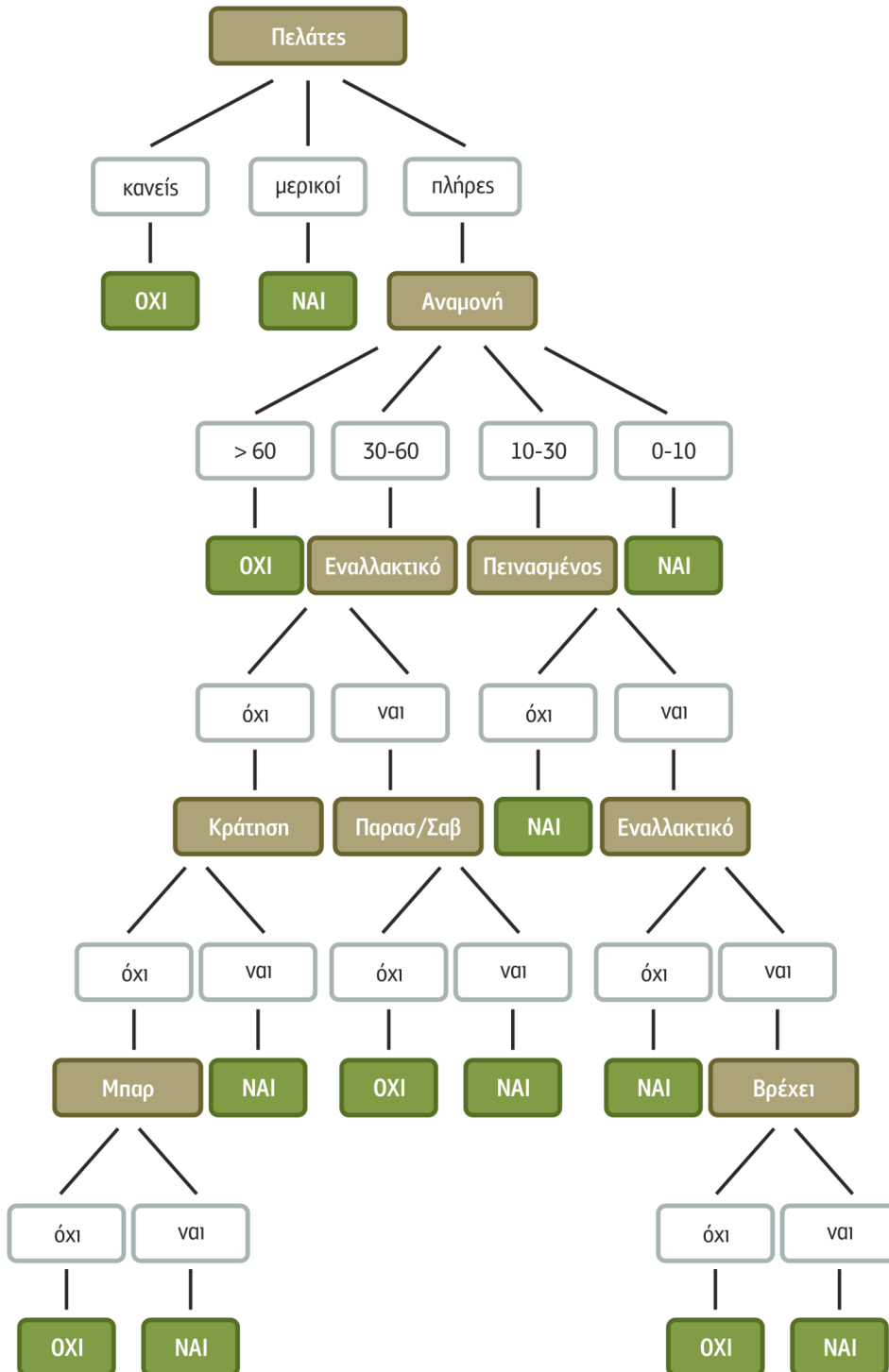
Το δένδρο που θα προκύψει θα είναι αυτό του σχήματος 4.12.



Σχήμα 4.12 Δένδρο απόφασης για το πρόβλημα αναμονής σε εστιατόριο

Ανάλογα με το ευρετικό κριτήριο που έχει, ο ID3 θα δημιουργήσει και διαφορετικό ΔΑ. Αν, για παράδειγμα, το κριτήριο είναι να εφαρμοστούν όσο γίνεται περισσότερα χαρακτηριστικά, για να αποφευχθεί η πρόωρη σύγκλιση και να είναι πιο πλούσια η γνώση που θα προκύψει, πρέπει να επιλεγούν τα χαρακτηριστικά με διαφορετική σειρά.

Στο σχήμα 4.13 παρουσιάζεται ένα πιθανό ΔΑ για το πρόβλημα αναμονής σε εστιατόριο που λαμβάνει υπόψη του το συγκεκριμένο κριτήριο.

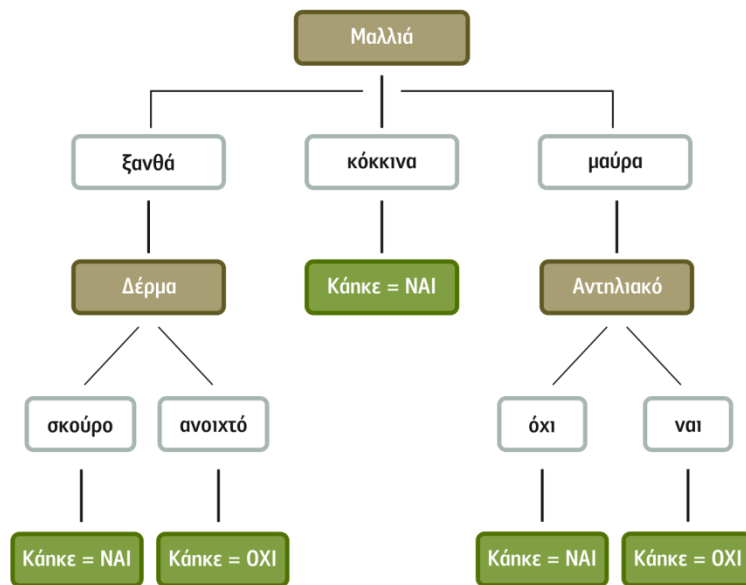


Σχήμα 4.13 Δένδρο απόφασης για το πρόβλημα αναμονής σε εστιατόριο που προσπαθεί να εξαντλήσει τα χαρακτηριστικά

Από δένδρα απόφασης σε κανόνες

Ένα δένδρο απόφασης μπορεί πολύ εύκολα να μετατραπεί σε ένα σύνολο κανόνων, απεικονίζοντας έναν προς έναν όλους τους κόμβους του δένδρου από τη ρίζα προς κάθε διαφορετικό κόμβο-φύλλο.

Για παράδειγμα, στο πρόβλημα της διαφήμισης αντηλιακού, το δένδρο απόφασης που δημιουργεί ο ID3 απεικονίζει 5 κανόνες (βλέπε Σχήμα.14).



R_1 : IF (Μαλλιά μαύρα) AND (Δέρμα σκούρο) THEN Κάπκε= ΟΧΙ
 R_2 : IF (Μαλλιά μαύρα) AND (Δέρμα ανοιχτό) THEN Κάπκε= ΝΑΙ
 R_3 : IF (Μαλλιά κόκκινα) THEN Κάπκε= ΝΑΙ
 R_4 : IF (Μαλλιά ξανθά) AND (Αντηλιακό ναι) THEN Κάπκε= ΟΧΙ
 R_5 : IF (Μαλλιά ξανθά) AND (Αντηλιακό όχι) THEN Κάπκε= ΝΑΙ

Σχήμα 4.14 Δένδρο απόφασης για το πρόβλημα της διαφήμισης αντηλιακού

Πλεονεκτήματα και Μειονεκτήματα των ΔΑ

Πλεονεκτήματα

- Φυσικός και κατανοητός τρόπος αναπαράστασης της γνώσης για αποσαφήνιση,
- Παραγόμενος αυτόματα εκτελέσιμος κώδικας, εφόσον κάθε δέντρο μεταφράζεται σε μια ακολουθία **if-then** εντολών,
- Αποτελεί παραδοσιακό μοντέλο μηχανικής μάθησης.

Μειονεκτήματα

- Υπάρχει δυσκολία στη μετατροπή αριθμητικών τιμών χαρακτηριστικών σε αντίστοιχα κατηγορήματα (για παράδειγμα μια τιμή του χαρακτηριστικού *Ηλικία* σε κατηγορίες 20-39, 40-59, και 60-79).
- Προκύπτει ανάγκη κλαδέματος κλαδιών που αναπτύσσονται συνεχώς. Η υπερβολική ανάπτυξη μειώνει μεν το σφάλμα στο σύνολο εκπαίδευσης, αλλά με κόστος την αύξηση του πλήθους των απαιτούμενων ελέγχων σφάλματος.

- Υπάρχει δυσχέρεια στη διαχείριση χαρακτηριστικών με πολλές τιμές
- Δυσκολεύεται ο αλγόριθμος να λειτουργήσει σωστά, όταν στο σύνολο εκπαίδευσης υπάρχουν δείγματα χωρίς τιμές σε ορισμένα από τα χαρακτηριστικά τους, σύνηθες φαινόμενο κατά τη συγκέντρωση και καταγραφή δειγμάτων.

4.4 Μάθηση με Συλλογιστική Βασισμένη σε Περιπτώσεις

Η **Συλλογιστική Βασισμένη σε Περιπτώσεις-ΣΒΠ** (Case-Based Reasoning-CBR) βασίζεται στην αξιοποίηση καταγεγραμμένης εμπειρίας και στην έρευνα περιπτώσεων προβλημάτων (cases) που αντιμετωπίστηκαν επιτυχώς στο παρελθόν, για να επιλυθεί με παραπλήσιο τρόπο ένα νέο πρόβλημα. Η επιτυχής έρευνα για την επιλογή κατάλληλων παρελθοντικών περιπτώσεων βασίζεται στην ομοιότητά τους με την τρέχουσα περίπτωση. Η ΣΒΠ/CBR ονομάζεται και **Συλλογιστική με Αναλογίες** (Analogical Reasoning) και χρησιμοποιείται (ίσως υποσυνείδητα) από τους περισσότερους ανθρώπους τόσο για επίλυση προβλημάτων όσο και για απόκτηση εμπειρίας.

Η επίλυση μιας νέας περίπτωσης προκύπτει από την ανάκτηση της πλέον σχετικής περίπτωσης από τις επιλεγμένες παρελθοντικές περιπτώσεις και την προσαρμογή της λύσης της. Η ΣΒΠ/CBR βασίζεται σε δυο αρχές σχετικές με τη φύση του κόσμου:

- Ο κόσμος είναι κανονικός: όμοια προβλήματα έχουν όμοια λύση.
- Προβλήματα που κάποιος αντιμετωπίζει τείνουν να ξανασυμβούν. Άρα, μελλοντικά προβλήματα είναι πολύ πιθανόν να μοιάζουν με τρέχοντα προβλήματα.

Η ΣΒΠ/CBR είναι από τις μεθόδους Μηχανικής Μάθησης που βασίζονται κατά κύριο λόγο στις ειδικές γνώσεις που έχουν αποκτηθεί κατά την επιτυχή αντιμετώπιση παρελθοντικών περιπτώσεων και έχουν καταχωρηθεί μέσα σε βάσεις περιπτώσεων με τρόπο που επιτρέπει την αξιοποίησή τους για την αντιμετώπιση παρόμοιων νέων περιπτώσεων.

Η ΣΒΠ/CBR χρησιμοποιείται σε δύο περιπτώσεις: για διερμηνεία (interpretation) και για επίλυση προβλημάτων (problem-solving). Τα συστήματα ΣΒΠ/CBR βρίσκουν εφαρμογή σε ποικίλους τομείς, όπως είναι η Νομική, όπου οι αποφάσεις του δικαστή βασίζονται στο δικαστικό προηγούμενο, και η Εγκληματολογία, είτε πρόκειται για κοινό έγκλημα, όπου για παράδειγμα τα αποτυπώματα και το DNA που υπάρχουν σε βάσεις δεδομένων βοηθούν στην ανακριτική διαδικασία μέσω της ταυτοποίησης, είτε πρόκειται για ηλεκτρονικό έγκλημα. Επίσης, η ΣΒΠ/CBR βρίσκει εφαρμογή σε συστήματα διαχείρισης επιχειρήσεων, οικονομικής διαχείρισης, διάγνωση σφαλμάτων και πρόβλεψης.

Στην ΤΝ, οι ρίζες του ΣΒΠ/CBR βρίσκονται στα έργα του Roger Schank (1982) για τη δυναμική μνήμη, τον σημαντικό ρόλο που παίζει η απομνημόνευση παρελθοντικών καταστάσεων (episodes, cases), τον τρόπο ανάκλησή τους για την επίλυση των προβλημάτων και την καινούρια γνώση που αποκτάται. Άλλες γνώσεις στο πεδίο των συστημάτων ΣΒΠ/CBR προέρχονται από τη μελέτη της αναλογικής σκέψης.

Η Συλλογιστική Βασισμένη σε Περιπτώσεις είναι ένας τρόπος επίλυσης προβλημάτων που, από πολλές απόψεις, είναι θεμελιωδώς διαφορετική από άλλες σημαντικές προσεγγίσεις της ΤΝ (Leake, 1996). Αντί να στηρίζεται αποκλειστικά σε γενικές γνώσεις ενός τομέα προβλημάτων, όπως ένα έμπειρο σύστημα, ή στις ενώσεις διαφορετικών τομέων, εδράζεται στις γενικευμένες σχέσεις μεταξύ των περιγραφών του προβλήματος και των συμπερασμάτων του.

Με την πρώτη ματιά, η ΣΒΠ/CBR μπορεί να φαίνεται παρόμοια με τους υπόλοιπους αλγόριθμους Επαγωγικής Μάθησης που χρησιμοποιούνται στη Μηχανική Μάθηση.

Όπως και σε έναν επαγωγικό αλγόριθμο μάθησης κανόνων, η ΣΒΠ/CBR ξεκινά με ένα σύνολο περιπτώσεων ή παραδειγμάτων εκπαίδευσης και σχηματίζει γενικεύσεις από τα παραδείγματα αυτά, αν και το πραγματοποιεί έμμεσα, εντοπίζοντας τα κοινά σημεία μεταξύ μιας ανακτηθείσας περίπτωσης και του προβλήματος-στόχου.

Βασική διαφορά, όμως, μεταξύ της ρητής γενίκευσης σε ΣΒΠ/CBR και της γενίκευσης από παραδείγματα για την εξαγωγή ενός κανόνα έγκειται στο πότε γίνεται η γενίκευση αυτή. Ένας κλασικός αλγόριθμος επαγωγής, όπως για παράδειγμα τα ΔΑ, αντλεί τις γενικεύσεις του από ένα σύνολο παραδειγμάτων εκπαίδευσης. Δηλαδή, εκτελεί μια βιαστική γενίκευση. Για παράδειγμα, αν στον αλγόριθμο

ID3 το σύνολο εκπαίδευσης περιέχει ως δείγματα τη μέθοδο παρασκευής μαρμελάδας νεράντζι, μαρμελάδας πορτοκάλι και μαρμελάδας λεμόνι, αναμένεται με τη λήξη της φάσης εκπαίδευσης να προκύψει ένα σύνολο γενικών κανόνων για την παραγωγή όλων των τύπων της παρασκευής μαρμελάδας, γενικεύοντας έτσι, σιωπηρά, το σύνολο των καταστάσεων στις οποίες μπορεί να χρησιμοποιηθεί η βασική μέθοδος παρασκευής μαρμελάδας. Μέχρι να χρησιμοποιηθεί ο κανόνας για έναν νέο τύπο μαρμελάδας, έστω μαρμελάδα φράουλα, δεν θα είναι γνωστό αν η γενίκευση που επιχειρήθηκε, για να παραχθεί ο κανόνας, ήταν η ορθή.

Αυτό έρχεται σε αντίθεση με την ΣΒΠ/CBR, η οποία καθυστερεί (σιωπηρά) τη γενίκευση των περιπτώσεων της μέχρι τη στιγμή οπότε θα δοθεί ο συγκεκριμένος στόχος. Αυτό αποκαλείται στρατηγική “ράθυμης γενίκευσης”. Στο παράδειγμα των μαρμελάδων, στη ΣΒΠ/CBR έχει ήδη δοθεί ο στόχος του προβλήματος που είναι η παρασκευή μαρμελάδας φράουλας. Έτσι, η μέθοδος μπορεί να γενικεύσει τις τρεις υπάρχουσες περιπτώσεις της ακριβώς όσο χρειάζεται, για να καλυφθεί αυτή η νέα περίπτωση.

Ως εκ τούτου, η ΣΒΠ/CBR τείνει να είναι μια καλή προσέγγιση για τομείς στους οποίους υπάρχουν πολλοί εναλλακτικοί τρόποι για να γενικευθεί μια υπόθεση και μας ενδιαφέρει να επιλέξουμε τον καταλληλότερο, όταν είναι γνωστός ο στόχος του προβλήματος που θέλουμε να λύσουμε.

4.4.1 Χαρακτηριστικά της ΣΒΠ/CBR

Ένα σύστημα ΣΒΠ/CBR αποτελείται από τα εξής βασικά τμήματα:

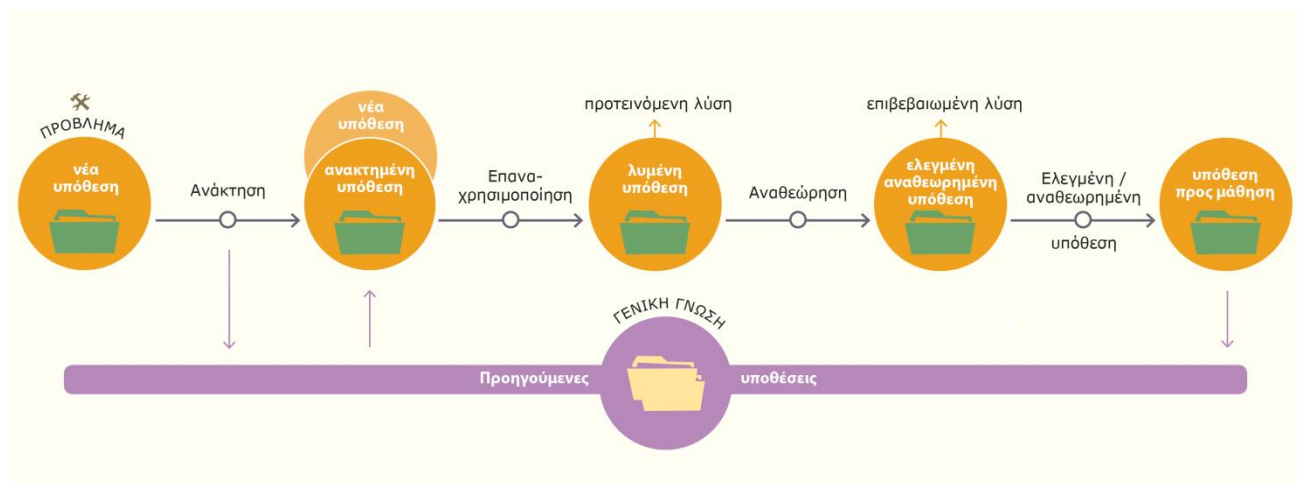
- μια **βιβλιοθήκη** από παλιές περιπτώσεις (case library),
- μια μέθοδο για το ταίριασμα και την **ανάκληση** από τη βιβλιοθήκη της περίπτωσης που είναι περισσότερο όμοια με την τωρινή (case retrieval) ως προς τα κυριότερα χαρακτηριστικά του προβλήματος.
- μια μέθοδο για την **προσαρμογή** της λύσης που δόθηκε στο παρελθόν (case adaptation), ώστε αυτή, με κάποιες τροποποιήσεις, να επαναχρησιμοποιηθεί,
- μια μέθοδο για τη δοκιμή, **επαλήθευση** και επιδιόρθωση της προσαρμοσμένης λύσης (case verification),
- μια μέθοδο για την **εκμάθηση** της λύσης (case learning), η οποία θα κρίνει αν η νέα περίπτωση και η προσαρμοσμένη λύση της συνιστούν μια πολύ διαφορετική περίπτωση από αυτές που βρίσκονται στη βιβλιοθήκη και πρέπει να προστεθούν σε αυτήν. Η διαδικασία της **εκμάθησης** καθιστά τα συστήματα ΣΒΠ/CBR συστήματα μηχανικής μάθησης.

4.4.2 Ο Κύκλος ΣΒΠ/CBR

Ένας κύκλος ΣΒΠ/CBR μπορεί να περιγραφεί με βάση τις εξής 4 λειτουργίες:

- **Ανάκτηση** (retrieve) της πιο όμοιας περίπτωσης ή περιπτώσεων,
- **Επαναχρησιμοποίηση** (reuse) των πληροφοριών και της γνώσης, στη συγκεκριμένη περίπτωση για να λυθεί το πρόβλημα,
- **Τροποποίηση** (revise) της προτεινόμενης λύσης ,
- **Διατήρηση** (retain) των τμημάτων αυτών της εμπειρίας που ενδέχεται να φανούν χρήσιμα για μελλοντική επίλυση προβλημάτων.

Ένα νέο πρόβλημα θα έχει λυθεί με **την ανάκτηση** μίας ή περισσότερων προηγούμενων περιπτώσεων, **την επαναχρησιμοποίηση της περίπτωσης** με τον ένα ή με τον άλλο τρόπο, την **τροποποίηση** της λύσης, που βασίζεται στην επαναχρησιμοποίηση μιας προηγούμενης περίπτωσης και τη **διατήρηση** της νέας εμπειρίας με την ενσωμάτωσή της στην υπάρχουσα βάση των περιπτώσεων. Καθεμία από τις τέσσερις λειτουργίες περιλαμβάνει μια σειρά από πιο συγκεκριμένες ενέργειες (Aamodt & Plaza, 1994). Στο σχήμα 4.15 φαίνεται αυτός ο κύκλος.



Σχήμα 4.15 Ο κύκλος των διεργασιών της Συλλογιστικής Βασισμένης σε Περιπτώσεις

Δείτε [κινούμενη εικόνα 4.2 - Ο κύκλος των διεργασιών της CBR](#)

Μια πρώτη περιγραφή του προβλήματος (στα αριστερά του σχήματος 4.15) καθορίζει μια νέα περίπτωση (New Case). Η νέα περίπτωση χρησιμοποιείται για την ανάκτηση (RETRIEVE) μιας περίπτωσης από τη συλλογή των προηγούμενων περιπτώσεων (Previous Cases). Η ανακτημένη περίπτωση (Retrieved Case) συνδυάζεται με τη νέα περίπτωση-μέσω της επαναχρησιμοποίησης (REUSE) ως προτεινόμενη λύση (Suggested Solution) στο υπό επίλυση πρόβλημα. Μέσω της διαδικασίας αναθεώρησης (REVISE), η λύση που προτείνεται (Solved Case) δοκιμάζεται στο πραγματικό περιβάλλον ή αξιολογείται από έναν ειδικό (Tested Case) και, εφόσον επιτύχει, γίνεται αποδεκτή (Confirmed Solution). Διαφορετικά, τροποποιείται κατάλληλα (Repaired Case). και επαναξιολογείται, έως ότου γίνει αποδεκτή. Κατά τη διαδικασία διατήρησης (RETAIN), η χρήσιμη εμπειρία (Learned Case) διατηρείται για μελλοντική χρήση και η βάση των προηγούμενων περιπτώσεων ενημερώνεται είτε με τη νέα επιλυμένη περίπτωση είτε με τη μετατροπή της ανακτημένης περίπτωσης, για να αποτελέσει λύση.

Όπως παρουσιάζεται στο σχήμα 4.15, η γενική γνώση συνήθως επιτελεί σημαντικό ρόλο σε αυτό τον κύκλο, υποστηρίζοντας τις διαδικασίες της ΣΒΠ/CBR. Η υποστήριξη αυτή μπορεί να είναι πολύ αδύναμη (ή να μην υπάρχει) ή πολύ ισχυρή, ανάλογα με το είδος της μεθόδου ΣΒΠ/CBR. Με τον όρο *γενική γνώση* εννοείται η γνώση που εξαρτάται από το γενικό τομέα, σε αντίθεση με την *ειδική γνώση* που ενσωματώνεται από τις περιπτώσεις.

Ο γενικός αλγόριθμος της ΣΒΠ/CBR μπορεί να συνοψιστεί ως:

ΑΡΧΗ

δέξου τις προδιαγραφές του νέου προβλήματος
προσδιόρισε τα χαρακτηριστικά αναζήτησης μέσα στη βάση
περιπτώσεων,
ανάκτησε ένα σύνολο περιπτώσεων με χαρακτηριστικά που
ταιριάζουν,
επίλεξε μια περίπτωση.
Επανάλαβε
τροποποίησε την περίπτωση,
αξιολόγησε τη λύση που δίνει.

Έως ότου η λύση είναι ικανοποιητική

ΤΕΛΟΣ

Για τη δεικτοδότηση των περιπτώσεων στη μνήμη, οι προδιαγραφές ενός καινούριου προβλήματος μετατρέπονται σε πρότυπο προς αντιστοίχιση. Το πρότυπο στη συνέχεια μπορεί να χρησιμοποιηθεί ως έχει ή μπορεί να τροποποιηθεί από το χρήστη. Το πρότυπο μπορεί να συγκριθεί με κάθε περίπτωση στη μνήμη ή αυτό μπορεί να παρέχει ένα σύνολο από δείκτες, ώστε να περιοριστεί ο χώρος αναζήτησης και μόνο ένα υποσύνολο σχετικών περιπτώσεων να συγκριθούν μαζί του. Η ανάκτηση μπορεί να βασίζεται σε μια τέλεια αντιστοίχιση ή σε μερικές αντιστοιχίσεις.

Το πρότυπο αποτελείται από ένα σύνολο σταθμισμένων χαρακτηριστικών αντιστοίχισης και η κάθε περίπτωση που έχει ανακτηθεί είναι δεικτοδοτημένη σύμφωνα με αυτά, έτσι ώστε η επιλογή να βασίζεται στην ανακτηθείσα περίπτωση με την πιο ικανοποιητική ταύτιση. Δηλαδή, αν το πρότυπο αποτελείται από ένα σύνολο χαρακτηριστικών, η επιλογή βασίζεται στην περίπτωση που έχει τα περισσότερα κοινά χαρακτηριστικά με αυτό (Maher κ.ά., 1995, σ.4).

Η διαδικασία της τροποποίησης προσαρμόζει μια προηγούμενη περίπτωση σε ένα νέο πρόβλημα συνδέοντας τύπους με τα χαρακτηριστικά ή τις παραμέτρους που μπορούν να αλλάξουν. Όταν προσδιορίζεται το περιεχόμενο της μνήμης, μερικά χαρακτηριστικά αναγνωρίζονται ως τροποποιήσιμα. Καθένα από τα χαρακτηριστικά έχει έναν τύπο ή μια διαδικασία που αξιολογείται κατά τη διάρκεια της τροποποίησης.

Όταν μια ανακτημένη περίπτωση επιλεγεί ως βάση για τη νέα λύση, υπάρχουν περιορισμοί που παρέχουν τη γνώση που απαιτείται για τον έλεγχο των διαφορών μεταξύ της επιλεγμένης περίπτωσης και του νέου προβλήματος.

4.4.3 Κύριοι τύποι μεθόδων ΣΒΠ/CBR

Ο μηχανισμός της ΣΒΠ/CBR καλύπτει ένα εύρος διαφορετικών μεθόδων για την οργάνωση, την ανάκτηση, την αξιοποίηση και τη δεικτοδότηση των γνώσεων που διατηρούνται από παρελθοντικές περιπτώσεις. Οι περιπτώσεις διατηρούνται είτε ως συγκεκριμένες εμπειρίες είτε ως σύνολο παρόμοιων περιπτώσεων που μπορεί να αποτελέσουν μια γενικευμένη περίπτωση. Η λύση από μία προηγούμενη περίπτωση μπορεί να εφαρμοστεί άμεσα στο σημερινό πρόβλημα ή να τροποποιηθεί ανάλογα με τις διαφορές μεταξύ των δύο περιπτώσεων. Μερικές μέθοδοι ΣΒΠ/CBR χρησιμοποιούν ένα αρκετά μεγάλο μέρος των δεικτοδοτημένων περιπτώσεων μέσα στις βάσεις, παλαιών περιπτώσεων ενώ άλλες βασίζονται σε ένα πιο περιορισμένο σύνολο χαρακτηριστικών περιπτώσεων. Προηγούμενες περιπτώσεις μπορεί να ανακτηθούν και να αξιολογηθούν διαδοχικά ή παράλληλα. Οι μέθοδοι ΣΒΠ/CBR μπορούν να είναι καθαρά αυτόνομες και αυτόματες ή να αλληλεπιδρούν σε μεγάλο βαθμό με τον χρήστη για την υποστήριξη και την καθοδήγηση των επιλογών του.

Τρεις είναι οι χαρακτηριστικοί σχετικοί τύποι ΣΒΠ/CBR σε σχέση με τη δομή των περιπτώσεων μέσα στη βάση:

- Διαρθρωτικός (διαθέτει κοινό δομημένο λεξιλόγιο, δηλαδή μια οντολογία)
- Κειμένου (οι περιπτώσεις παρουσιάζονται ως ελεύθερο κείμενο, δηλαδή ως συμβολοσειρές)
- Διαλόγου (μια περίπτωση αντιπροσωπεύεται από μια λίστα ερωτήσεων που ποικίλλει από τη μια περίπτωση στην άλλη, η δε γνώση αποκτάται από συζητήσεις μεταξύ συστήματος και χρήστη).

Στην πραγματικότητα, ο όρος “Συλλογιστική Βασισμένη σε Περιπτώσεις –ΣΒΠ/CBR” είναι ένας μόνο από τους όρους που χρησιμοποιούνται όσον αφορά τα συστήματα αυτού του είδους.

Άλλοι όροι που σχετίζονται με την ΣΒΠ/CBR παρατίθεται στη συνέχεια (Aamodt&Plaza, 1994):

- Συλλογιστική Βασισμένη σε Υποδείγματα (Exemplar-based reasoning)
- Συλλογιστική Βασισμένη σε Περιστατικά (Instance-based reasoning)
- Συλλογιστική Βασισμένη στη Μνήμη (Memory-based reasoning)
- Συλλογιστική Βασισμένη στη Αναλογική Σκέψη (Analogy-based reasoning)

Παρά το γεγονός ότι η ΣΒΠ/CBR χρησιμοποιείται ως γενικός όρος, η τυπική μέθοδος ΣΒΠ/CBR έχει κάποια χαρακτηριστικά που τη διακρίνει από τους άλλους τύπους που αναφέρονται εδώ. Πρώτον, μια τυπική περίπτωση συνήθως έχει κάποιο βαθμό πλούτου πληροφοριών που περιέχονται σε αυτήν, καθώς και μια σχετική *πολυπλοκότητα* όσον αφορά την οργάνωση του εσωτερικού της. Δηλαδή, ένα διάγραμμα χαρακτηριστικών γνωρισμάτων που κατέχουν κάποιες αξίες και μια αντίστοιχη κλάση δεν είναι αυτό που θα χαρακτηριζόταν ως μια τυπική περιγραφή της περίπτωσης. Η τυπική βασισμένη σε περιπτώσεις μέθοδος έχει ακόμα άλλο ένα χαρακτηριστικό: τη δυνατότητα να *τροποποιήσει*, να *προσαρμόσει* και να *ανακτήσει* μια λύση, όταν εφαρμόζεται σε ένα διαφορετικό πλαίσιο επίλυσης προβλημάτων.

4.4.4 Πλεονεκτήματα - Μειονεκτήματα

Τα πλεονεκτήματα της ΣΒΠ/CBR σε σχέση με τα κλασικά συστήματα διαχείρισης γνώσης (π.χ. συστήματα κανόνων) μπορούν να συνοψιστούν στα ακόλουθα:

Η ΣΒΠ/CBR βρίσκεται πιο κοντά στον τρόπο με τον οποίο σκέφτονται οι άνθρωποι. Τα συστήματα κανόνων βασίζονται στην εμπειρία που εκμαieύτηκε από κάποιον ειδικό, αλλά η γνώση του ειδικού αναπαρίσταται στον υπολογιστή πολύ διαφορετικά. Με άλλα λόγια, στα έμπειρα συστήματα η γνώση του ειδικού έχει υποστεί μετατροπές, είτε από τον ίδιο τον ειδικό που αναγκάζεται να ομαδοποιήσει τις εμπειρίες του ή από τον μηχανικό γνώσης ο οποίος πρέπει αφαιρετικά να βρει κοινά στοιχεία στις εμπειρίες του ειδικού και να τις μετατρέψει σε κανόνες, ενώ στη ΣΒΠ/CBR η εμπειρία του ειδικού αναπαρίσταται ευθέως με τη μορφή περιπτώσεων που αντιμετώπισε στο παρελθόν και ανακαλεί από τη μνήμη του. Τα προβλήματα που αντιμετωπίστηκαν στο παρελθόν συγκρίνονται με τις τωρινές καταστάσεις και οι λύσεις που υιοθετήθηκαν ξαναχρησιμοποιούνται με μικρές μετατροπές.

Η διαδικασία απόκτησης της γνώσης απλουστεύεται, γιατί τις περισσότερες φορές η γνώση των παρελθοντικών περιπτώσεων υπάρχει ήδη κάπου αποθηκευμένη, στην καλύτερη περίπτωση μέσα σε βάσεις δεδομένων που διατηρούν εταιρείες και οργανισμοί. Εάν δεν υπάρχει βιβλιοθήκη περιπτώσεων, τότε κάποιος ειδικός καλείται να τη δημιουργήσει από τις εμπειρίες που υπάρχουν διαθέσιμες για το πρόβλημα σε μια ποικιλία μορφών. Βέβαια, η περίπτωση αυτή απαιτεί πολύ μεγαλύτερο κόπο εκ μέρους του ειδικού από μια απλή συμμετοχή του στη διαδικασία εκμαieυσης γνώσης, όπως γίνεται στα έμπειρα συστήματα.

Το κυριότερο μειονέκτημα της συλλογιστικής των περιπτώσεων, όπως και κάθε άλλου πληροφορικού συστήματος, είναι η δυσχέρεια στην απόκτηση γνώσης για το πεδίο και το υπολογιστικό κόστος της αναζήτησης στη βιβλιοθήκη των περιπτώσεων. Επίσης, βασικό μειονέκτημα είναι οι δυσκολίες στην προσαρμογή της λύσης μιας ανακτηθείσας περίπτωσης στην τρέχουσα κατάσταση.

Στη συνέχεια, αναφέρονται συνοπτικά τα πλεονεκτήματα και τα μειονεκτήματα της ΣΒΠ/CBR:

Πλεονεκτήματα

- Επιτρέπει στο σύστημα να προτείνει λύσεις σε προβλήματα γρήγορα, αποφεύγοντας τον απαιτούμενο χρόνο της παραγωγής των απαντήσεων εξαρχής σε τομείς που δεν έχουν πλήρως αναλυθεί.
- Παρέχει στο σύστημα τα μέσα για να αξιολογεί τις λύσεις, όταν δεν υπάρχει διαθέσιμη αλγοριθμική μέθοδος.
- Βοηθάει στη διερμηνευση ορισμένων “ανοιχτών” και με λανθασμένο τρόπο διατυπωμένων προβλημάτων.
- Βοηθάει το σύστημα να εντοπίσει τα πιο σημαντικά χαρακτηριστικά του προβλήματος και να επικεντρώσει τη συλλογιστική του σε αυτά.
- Βοηθάει το σύστημα να αποφύγει την επανάληψη παρελθοντικών προβλημάτων τα οποία μπορεί να αναδείξουν οι περιπτώσεις.

Μειονεκτήματα

- Οι παρελθοντικές περιπτώσεις μπορεί να είναι “ελλιπείς”.

- Η βιβλιοθήκη των περιπτώσεων μπορεί να αναφέρεται σε ένα επιλυμένο πρόβλημα χωρίς να έχει συγκεντρωθεί ένα ικανοποιητικός αριθμός παρελθοντικών περιπτώσεων.
- Υπάρχει δυσκολία στην ανάκτηση των πιο σχετικών περιπτώσεων.

Η ΣΒΠ/CBR βρίσκει εφαρμογή σε μια ευρεία ποικιλία έργων επίλυσης προβλημάτων, συμπεριλαμβανομένων του καθορισμού στόχων (*planning*), της διάγνωσης (*diagnosis*) και του σχεδιασμού (*design*) ενός προβλήματος. Σε καθένα από αυτά τα έργα, οι περιπτώσεις είναι χρήσιμες στο να προτείνουν λύσεις, καθώς και στην προειδοποίηση πιθανών προβλημάτων που μπορεί να προκύψουν. Η ΣΒΠ/CBR βρίσκει επιπλέον εφαρμογή σε προβλήματα που απαιτούν ικανότητες διερμηνείας της διατιθέμενης γνώσης, όπως είναι η κατάταξη μιας νέας κατάστασης ενός προβλήματος, η απόδειξη της ορθότητας ενός επιχειρήματος ή η πρόβλεψη των αποτελεσμάτων μιας λύσης. Στην περίπτωση αυτή, η ΣΒΠ/CBR καλείται **Διερμηνευτική ΣΒΠ/CBR** (Interpretive CBR).

4.4.5 Δυσχέρεια στην απόκτηση γνώσης

Η απόκτηση γνώσης είναι η μετατροπή της προϋπάρχουσας εμπειρίας σε μια μορφή αναγνωρίσιμη και επεξεργάσιμη από ένα Βασισμένο στη Γνώση Σύστημα (Knowledge-Based System-KBS). Το πρόβλημα της δυσχέρειας στην απόκτηση γνώσης στα συστήματα ΣΒΠ/CBR είναι άρρηκτα συνδεδεμένο με το μέγεθος της βιβλιοθήκης και με την ποικιλία των αποθηκευμένων περιπτώσεων σε αυτή.

Συγκεκριμένα, σε ένα σύστημα θα πρέπει να ελέγχονται τα νέα στιγμιότυπα, προτού αυτά αποθηκευτούν, και να μην επιτρέπεται να καταχωρούνται παρόμοια στιγμιότυπα, αλλά να τροποποιούνται σε ένα, το οποίο θα δίνει λύση σε όλες τις παραπλήσιες περιπτώσεις. Απώτερος σκοπός της παραπάνω διαδικασίας είναι να μη δημιουργούνται δαιδαλώδεις βιβλιοθήκες περιπτώσεων καθιστώντας τις μη λειτουργικές και κατ' επέκταση επιβαρύνοντας το σύστημα.

Ένα άλλο πρόβλημα δημιουργείται, όταν μία περίπτωση αντιμετωπίζεται για πρώτη φορά από το σύστημα. Για παράδειγμα, όταν στο περιεχόμενο της βιβλιοθήκης δεν υπάρχει κάποια περίπτωση παρόμοια με την τρέχουσα, τότε η λύση πρέπει να δομηθεί από την αρχή. Για να αποκτήσει το σύστημα την απαιτούμενη γνώση, θα απαιτηθεί περισσότερος υπολογιστικός φόρτος και μια αρκετά χρονοβόρα διαδικασία. Εντούτοις, το κόστος είναι απαραίτητο, διότι διαφορετικά το σύστημα θα αναλώνεται στην τροποποίηση των ήδη υπάρχουσών περιπτώσεων, καταλήγοντας τελικά σε μια διαφορετική λύση από αυτήν που απαιτείται.

4.5 Νευρωνικά Δίκτυα

4.5.1 Γενικά

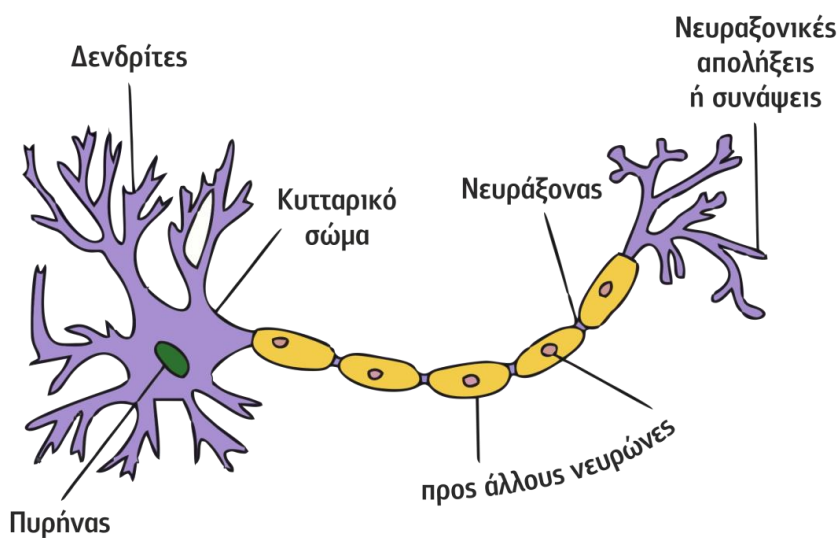
Ο όρος **Νευρωνικά Δίκτυα** (Neural Networks, Connectionist Networks, Parallel Distributed Processing Models) περιγράφει έναν αριθμό από διαφορετικά μαθηματικά μοντέλα, εμπνευσμένα από αντίστοιχα βιολογικά μοντέλα, δηλαδή μοντέλα που προσπαθούν να μιμηθούν τη συμπεριφορά των νευρώνων του ανθρώπινου εγκεφάλου.

Ήδη από τον 19ο αιώνα οι επιστήμονες παραδέχονται ότι ο εγκέφαλος αποτελείται από διακριτά στοιχεία, τους **νευρώνες** (neurons), που επικοινωνούν το ένα με το άλλο. Οι νευρώνες συνιστούν το βασικό δομικό κομμάτι του ανθρώπινου εγκεφάλου. Υπολογίζεται ότι ο εγκέφαλος περιέχει 10 δισ. περίπου νευρώνες τοποθετημένους σε ομάδες, καθεμία από τις οποίες συνιστά ένα φυσικό νευρωνικό δίκτυο. Έτσι, ο ανθρώπινος εγκέφαλος περιέχει εκατοντάδες φυσικά νευρωνικά δίκτυα, καθένα από τα οποία περιέχει χιλιάδες διασυνδεδεμένους νευρώνες με μέσο αριθμό διασυνδέσεων ανά νευρώνα 1000 με 10.000.



Εικόνα 4.1 [Μικροσκοπική φωτογραφία φυσικών νευρώνων](#)

Ένας νευρώνας διαχωρίζεται από τα υπόλοιπα κύτταρα με μια μεμβράνη και έχει την ικανότητα να μεταφέρει ηλεκτρικά σήματα από το νευρώνα αυτόν προς τους υπόλοιπους νευρώνες με τους οποίους επικοινωνεί.

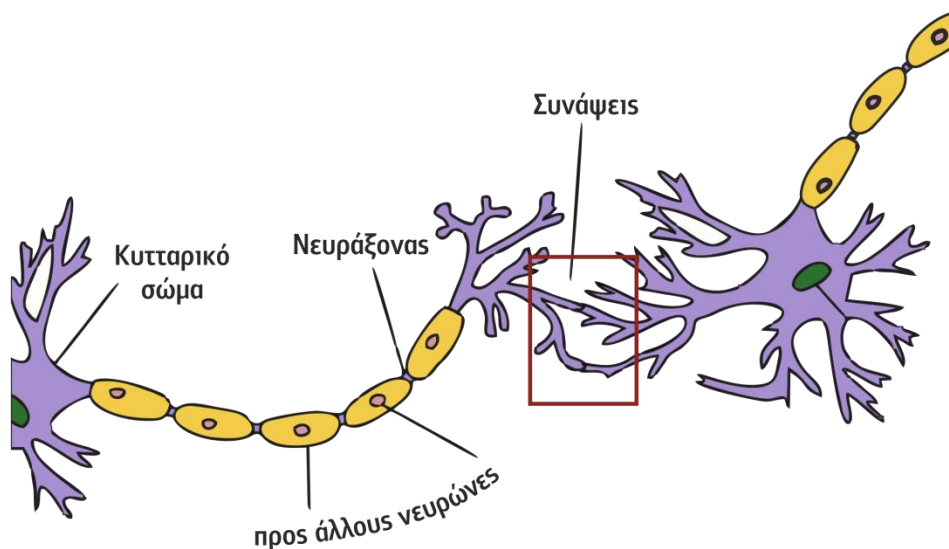


Σχήμα 4.16 Σχηματικό διάγραμμα ενός τυπικού νευρώνα

Κάθε νευρώνας αποτελείται από 3 κύρια τμήματα (βλέπε Σχήμα 4.16):

- τους **δενδρίτες** (dendrites), οι οποίοι λειτουργούν ως κανάλια εισόδου για το νευρώνα,
- το κυρίως κυτταρικό σώμα (cell body),
- τον άξονα του κυττάρου-νευροάξονα (axon), που συνδέει ένα νευρώνα με άλλους νευρώνες.

Ο άξονας του ενός νευρώνα μεταφέρει σήματα στους δενδρίτες γειτονικών νευρώνων μέσω του σημείου ένωσης που ονομάζεται νευροαξονική απόληξη ή σύναψη (synapse). Ένας νευρώνας μπορεί να λάβει σήματα από ένα σύνολο γειτονικών νευρώνων μέσω των δενδριτών, να τα επεξεργαστεί και να τροφοδοτήσει την έξοδό του μέσω του άξονα προς ένα άλλο σύνολο γειτονικών νευρώνων (βλέπε Σχήμα 4.17). Τα σήματα που έρχονται μέσω των δενδριτών “ζυγίζονται” και τα αποτελέσματα αθροίζονται. Όταν το άθροισμα ξεπεράσει το οριακό επίπεδο (τιμή κατωφλίου), ο νευρώνας δημιουργεί μια έξοδο (με τη μορφή νευρικής ώσης ή ηλεκτρικού σήματος) στον άξονά του, η οποία εν συνεχεία μέσω των συνάψεων θα μεταφερθεί στους γειτονικούς νευρώνες.



Σχήμα 4.17 Φυσικοί διασυνδεδεμένοι νευρώνες

Για την παραγωγή σήματος, ο νευρώνας δέχεται σήματα εισόδου που επιδρούν στο δυναμικό του αυξομειώνοντάς το. Όταν αθροιστικά το δυναμικό ξεπεράσει κάποιο όριο (ποικίλλει από κατηγορία σε κατηγορία κυττάρου μεταξύ -40 mV και -75 mV), τότε ο νευρώνας διεγείρεται και παράγει το ηλεκτρικό σήμα. Ο νευρώνας μεταφέρει το ηλεκτρικό σήμα πάντοτε προς μια προβλέψιμη και σταθερή κατεύθυνση. Υπάρχουν δυο διακριτές καταστάσεις σημάτων:

- Δυναμικό ηρεμίας
- Δυναμικό ενέργειας

Τα σήματα που λαμβάνονται από ένα νευρώνα μεταβάλλονται από τα ηλεκτρικά χαρακτηριστικά των επαφών των συνάψεων, ώστε να εμποδίζονται μερικά και να επιτρέπεται σε άλλα να διαδοθούν. Τα ηλεκτρικά χαρακτηριστικά των συνάψεων αποτελούν κάποιο είδος πληροφορίας μοναδικής σε κάθε νευρώνα. Με αυτό τον τρόπο οι πληροφορίες που κρατούνται από ένα δίκτυο κατανέμονται στους νευρώνες του.

Η μεταβίβαση πληροφορίας γίνεται με βάση το δυναμικό ενέργειας που καθορίζεται όχι από τον τύπο του σήματος, αλλά από την οδό του εγκεφάλου μέσα από διακριτά επικοινωνούντες νευρώνες από τους οποίους περνάει το σήμα.

Η έκφραση “διακριτά επικοινωνούντα στοιχεία” είναι η βάση του ορισμού των ψηφιακών κυκλωμάτων, αλλά σε καμία περίπτωση ο εγκέφαλος δεν είναι ένας ψηφιακός ηλεκτρονικός υπολογιστής

ούτε ένας ηλεκτρονικός υπολογιστής μπορεί να αντικαταστήσει τον εγκέφαλο. Μία από τις βασικές αιτίες είναι ότι τα πολλαπλά φυσικά νευρωνικά δίκτυα του εγκεφάλου είναι οργανωμένα σε τμήματα που λειτουργούν παράλληλα· το καθένα από αυτά μπορεί να προκαλέσει ανεξάρτητες συμπεριφορές, εμφανίζοντας πλαστικότητα στην ανάληψη λειτουργιών (δηλαδή προσαρμόζονται σε αλλαγές που επιφέρει το εσωτερικό ή εξωτερικό περιβάλλον, ώστε να εξακολουθούν να λειτουργούν επιτυχώς στο μέτρο του δυνατού) και ως εκ τούτου δεν μπορούν να εξομοιωθούν με ηλεκτρονικά κυκλώματα που δε διαθέτουν παρόμοια χαρακτηριστικά.

4.5.2 Προσομοίωση φυσικών νευρωνικών δικτύων με τεχνητά νευρωνικά δίκτυα

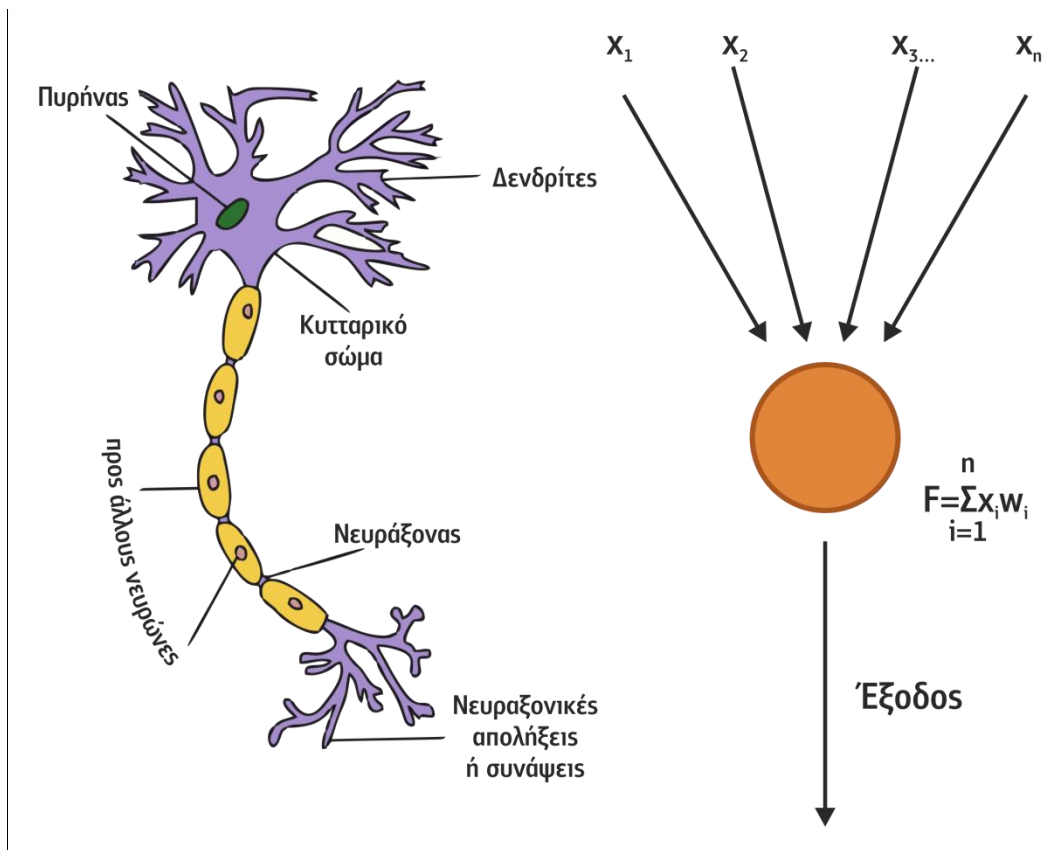
Τα μαθηματικά μοντέλα των τεχνητών νευρωνικών δικτύων, σε πλήρη αντιστοιχία με τα βιολογικά, αποτελούνται από έναν αριθμό απλών και με υψηλό βαθμό εσωτερικής διασύνδεσης επεξεργαστικών μονάδων, οργανωμένων σε στρώματα. Τα **Τεχνητά Νευρωνικά Δίκτυα-TND** (Artificial Neural Networks-ANN) επεξεργάζονται πληροφορίες ανταποκρινόμενα δυναμικά σε εξωτερικά ερεθίσματα (εισόδους). Κάθε **τεχνητός νευρώνας** αποτελείται από πολλές εισόδους x_i και μία μόνο έξοδο y . Κάθε είσοδος x_i “ζυγίζεται” με ένα βάρος w_i και τα αποτελέσματα αθροίζονται μέσω της **συνάρτησης αθροίσματος** (summation function) F :

$$F = \sum_i^n x_i w_i$$

Ο τεχνητός νευρώνας δίνει έξοδο μέσω της **συνάρτησης μετάβασης** (transfer function), μόνο όταν το ζυγισμένο άθροισμα των εισόδων είναι μεγαλύτερο μιας ορισμένης **τιμής κατωφλίου** (threshold value) θ , δηλαδή όταν:

$$\sum_i^n x_i w_i - \theta > 0$$

Ένας τεχνητός νευρώνας αποτελεί απλοποιημένο μοντέλο του φυσικού νευρώνα κατά το ότι τα βάρη διασύνδεσης σχηματίζουν τα ηλεκτρικά χαρακτηριστικά της επαφής της σύναψης και η τιμή κατωφλίου προσομοιώνει τη συμπεριφορά κορεσμού του φυσικού νευρώνα (βλέπε Σχήμα 4.18).



Σχήμα 4.18 Ο φυσικός νευρώνας σε σχέση με τον στοιχειώδη τεχνητό νευρώνα (Perceptron)

Ένα από τα απλούστερα ΤΝΔ που προσομοιώνουν τον φυσικό νευρώνα είναι ο **στοιχειώδης Perceptron** (basic Perceptron), δηλαδή ένα ΤΝΔ που αποτελείται από έναν μόνο νευρώνα. Η έξοδος a του Perceptron για ένα διάνυσμα εισόδου $x = (x_1, x_2, \dots, x_n)$ δίνεται μέσω της συνάρτησης μετάβασης g ως ακολούθως:

$$a = g \left(\sum_{i=1}^n x_i w_i \right)$$

Συνοπτική περιγραφή ενός ΤΝΔ:

- Τα ΤΝΔ συνήθως οργανώνονται σε **επίπεδα** (layers) τα οποία καλούνται και στρώματα. Τα ενδιάμεσα επίπεδα καλούνται κρυμμένα επίπεδα (hidden layers) και δεν είναι απαραίτητο να υπάρχουν.
- Τα επίπεδα αποτελούνται από έναν αριθμό **μονάδων** (units) ή κόμβων (nodes) που είναι έτσι συνδεδεμένες μεταξύ τους, ώστε μία μονάδα να έχει συνδέσμους με πολλές άλλες μονάδες του ίδιου ή άλλου επιπέδου.
- Οι μονάδες επιδρούν σε άλλες μονάδες με το να τις διεγείρουν ή να αναστέλλουν την ενεργοποίησή τους. Για να επιτευχθεί αυτό η μονάδα λαμβάνει το σταθμισμένο άθροισμα όλων των εισόδων μέσω των συνδέσμων που καταλήγουν σε αυτήν και παράγει μέσω της συνάρτησης μετάβασης μία μοναδική έξοδο, εάν το άθροισμα υπερβαίνει μία τιμή κατωφλίου.
- Οι εισοδοί παρουσιάζονται στο δίκτυο μέσω του επιπέδου εισόδου (input layer) το οποίο επικοινωνεί με έναν ή περισσότερα κρυμμένα επίπεδα. Τα κρυμμένα επίπεδα συνδέονται με το επίπεδο εξόδου (output layer) από το οποίο εξάγεται η απάντηση.

Βασικά στοιχεία της αρχιτεκτονικής των ΤΝΔ που πρέπει να καθοριστούν κατά τη δημιουργία τους είναι:

- Ο αριθμός των ενδιάμεσων κρυφών επιπέδων,
- Ο αριθμός των μονάδων (ή κόμβων) ανά επίπεδο,
- Ο τρόπος σύνδεσης των μονάδων μεταξύ τους,
- Η τιμή ενεργοποίησης (τιμή κατωφλίου),
- Η μορφή της συνάρτησης μετάβασης,
- Οι τιμές των αρχικών βαρών μεταξύ των μονάδων,
- Οι αλγόριθμοι (κανόνες εκπαίδευσης) που χρησιμοποιούνται, για να ενισχυθούν οι σύνδεσμοι μεταξύ των μονάδων κατά τη διαδικασία της εκπαίδευσης.

4.5.3 Ιστορική Αναδρομή

Τα ορόσημα στην εξέλιξη του χώρου των ΤΝΔ είναι τα ακόλουθα:

1943: McCulloch & Pitts – Δημιουργούν το πρώτο μοντέλο ΤΝΔ

1949: Hebb – Δημιουργεί το μοντέλο μάθησης που πήρε το όνομά του στο οποίο κάθε φορά που ενεργοποιείται μια σύναψη αυτή ενισχύεται, με αποτέλεσμα το δίκτυο να μαθαίνει “λίγο περισσότερο” το πρότυπο που του παρουσιάζεται εκείνη τη στιγμή.

1957: Rosenblatt- Προτείνει το στοιχειώδες ΤΝΔ του απλού αισθητήρα που ονόμασε Perceptron.

1969: Minsky & Papert- Αποδεικνύουν μαθηματικά ότι τα ΤΝΔ ενός επιπέδου δεν μπορούν να λύσουν μη γραμμικά προβλήματα.

1982: Μαθηματική απόδειξη ότι ένα ΤΝΔ πολλών επιπέδων μπορεί να αποθηκεύσει οποιαδήποτε πληροφορία.

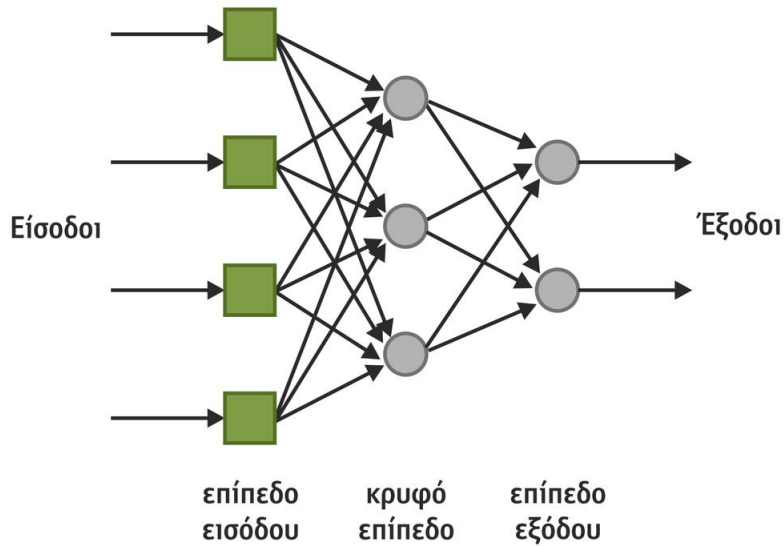
1986: Werbos & Rumelhart – Προτείνουν τη μέθοδο οπισθοδιάδοσης (backpropagation) για την εκπαίδευση ΤΝΔ.

4.5.4 Αρχιτεκτονική ΤΝΔ

Όσον αφορά το πώς είναι συνδεδεμένες οι μονάδες μεταξύ τους, υπάρχουν δυο βασικές κατηγορίες ΤΝΔ:

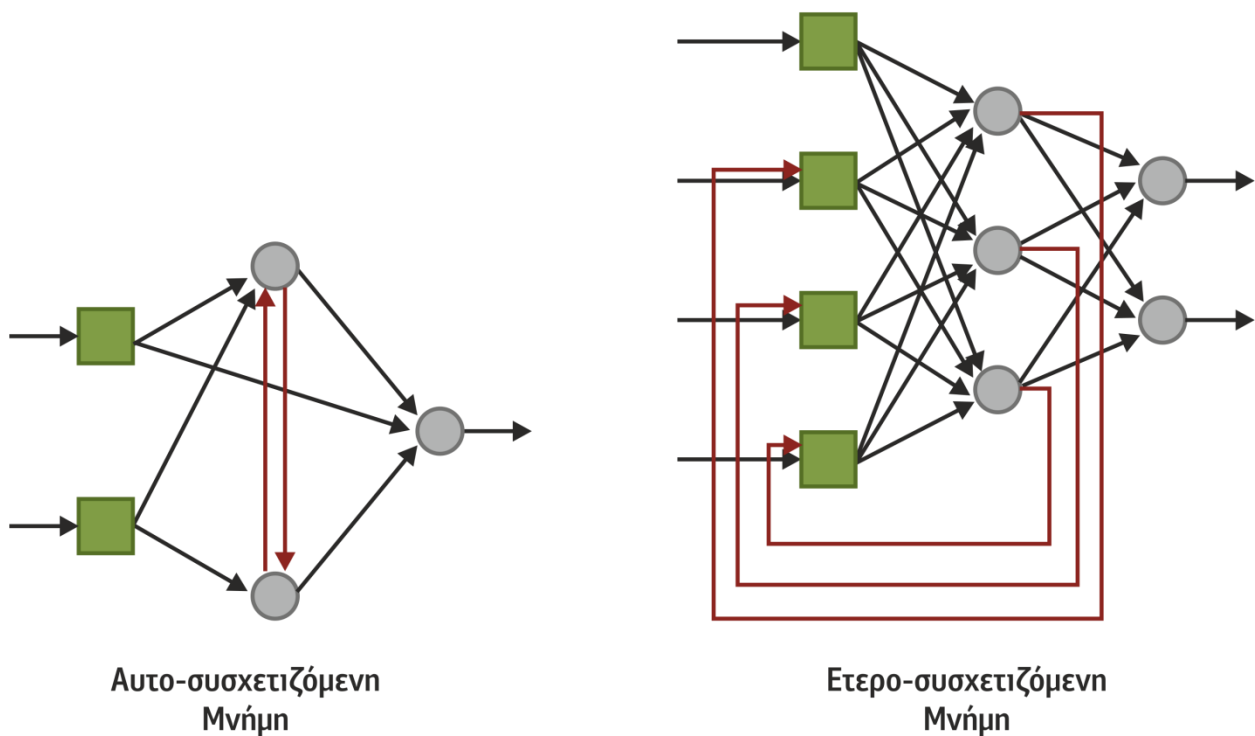
- **πρόσθιας τροφοδότησης** (feed forward) και
- **οπίσθιας τροφοδότησης** (feed backward)

Στα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, οι μονάδες είναι οργανωμένες σε διαφορετικά επίπεδα, ώστε οι μονάδες του ενός επιπέδου να τροφοδοτούν τις μονάδες του επόμενου επιπέδου, έως ότου τροφοδοτηθούν και οι μονάδες του τελευταίου επιπέδου (βλέπε Σχήμα 4.19). Δηλαδή, δεν υπάρχει έξοδος μονάδας ενός επιπέδου που να αποτελεί είσοδο μονάδας του ίδιου ή προηγούμενων επιπέδου. Τέτοια ΤΝΔ είναι τα δίκτυα οπισθοδιάδοσης (backpropagation).



Σχήμα 4.19 Παράδειγμα ΤΝΔ πρόσθιας τροφοδότησης

Στα οπισθίως τροφοδοτούμενα δίκτυα, που καλούνται και **ανατροφοδοτούμενα ΤΝΔ** (recurrent ANN), επιτρέπεται στις μονάδες ενός επιπέδου να τροφοδοτούν και μονάδες του ίδιου επιπέδου ή και προηγούμενων επιπέδων. Αν η ανατροφοδότηση αφορά κόμβους στο ίδιο επίπεδο, τότε τα δίκτυα καλούνται **αυτοσυσχετιζόμενες μνήμες** (autoassociated memories) διαφορετικά, καλούνται **ετεροσυσχετιζόμενες μνήμες** (heteroassociated memories) (βλέπε Σχήμα 4.20).



Σχήμα 4.20 Παράδειγμα ανατροφοδοτούμενων ΤΝΔ

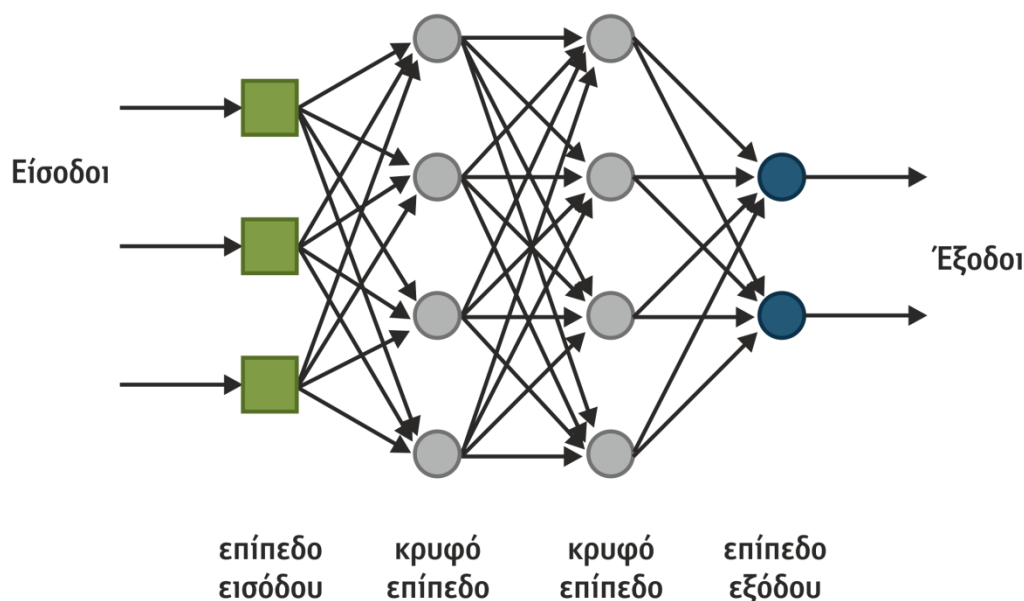
Στα ανατροφοδοτούμενα ΤΝΔ δεν υπάρχουν συνήθως άνω του ενός ενδιάμεσα (κρυφά) επίπεδα. Αν και τα ανατροφοδοτούμενα δίκτυα είναι πολύ χρήσιμα, τα περισσότερα των νευρωνικών δικτύων είναι πρόσθιας τροφοδότησης.

Πολυεπίπεδα ΤΝΔ

Κοινό χαρακτηριστικό της δομής των πολυεπίπεδων ΤΝΔ είναι ότι διαθέτουν τουλάχιστον ένα κρυφό επίπεδο. Οι κόμβοι των διάφορων επιπέδων μπορεί να είναι πλήρως συνδεδεμένοι (fully connected), δηλαδή κάθε κόμβος του ενός επιπέδου συνδέεται με όλους τους κόμβους του επόμενου, όπως στο [σχήμα 4.19](#), ή μερικώς συνδεδεμένοι (partially connected), όπως στο σχήμα 4.21.

Τα ΤΝΔ χαρακτηρίζονται, επιπλέον, με βάση τον τρόπο με τον οποίο είναι συνδεδεμένοι οι κόμβοι τους, όπως αναφέραμε στην προηγούμενη παράγραφο περί Αρχιτεκτονικής ΤΝΔ, δηλαδή αν είναι πρόσθιας τροφοδότησης (feed forward) ή οπίσθιας τροφοδότησης (feed backward ή recurrent).

Στην πλειοψηφία των εφαρμογών χρησιμοποιούνται δίκτυα πρόσθιας τροφοδότησης ενός κρυφού επιπέδου με πλήρως συνδεδεμένους κόμβους.

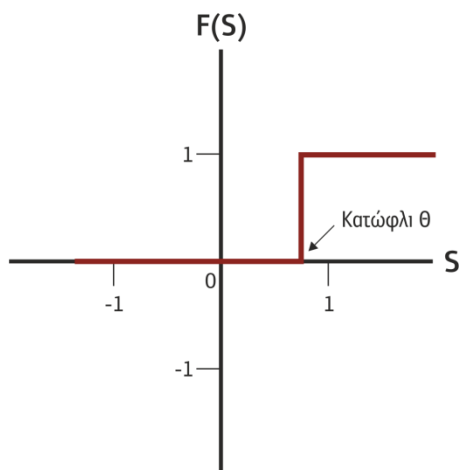


Σχήμα 4.21 Πολυεπίπεδο ΤΝΔ ενός κρυφού επιπέδου πρόσθιας τροφοδότησης 2 κρυμμένων επιπέδων με πλήρως συνδεδεμένους κόμβους

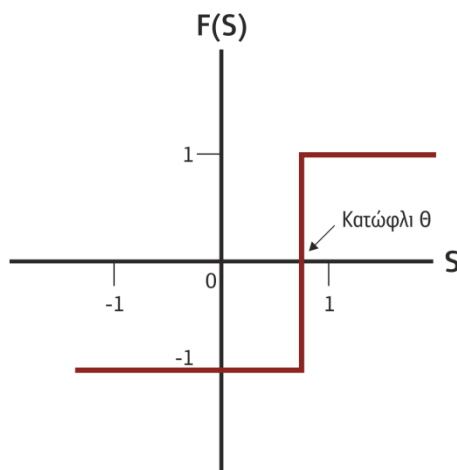
Συναρτήσεις μετάβασης

Όσον αφορά τις συναρτήσεις μετάβασης (transfer functions) οι πιο απλές είναι οι γραμμικές (βλέπε Σχήμα 4.22), όπως οι παρακάτω:

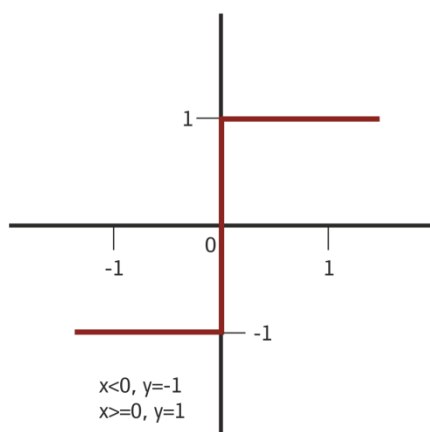
- **βηματικές συναρτήσεις ή συναρτήσεις κατωφλίου** (threshold functions),
 - **συναρτήσεις προσήμου** (sign functions),
 - **συναρτήσεις βηματικής μεταβολής** (hard limiter functions),
 - **συναρτήσεις αναρρίχησης** (ramping functions).
- κ.ά.



Βηματική (έξοδος 1 ή 0)

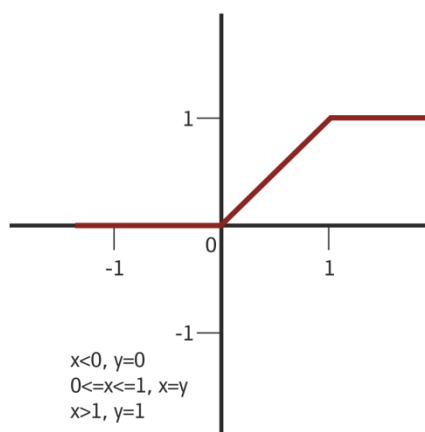


Πρόσημου
(έξοδος θετική ή αρνητική πληροφορία, +1, -1)



$$\begin{aligned} x < 0, & y = -1 \\ x \geq 0, & y = 1 \end{aligned}$$

Hard Limiter

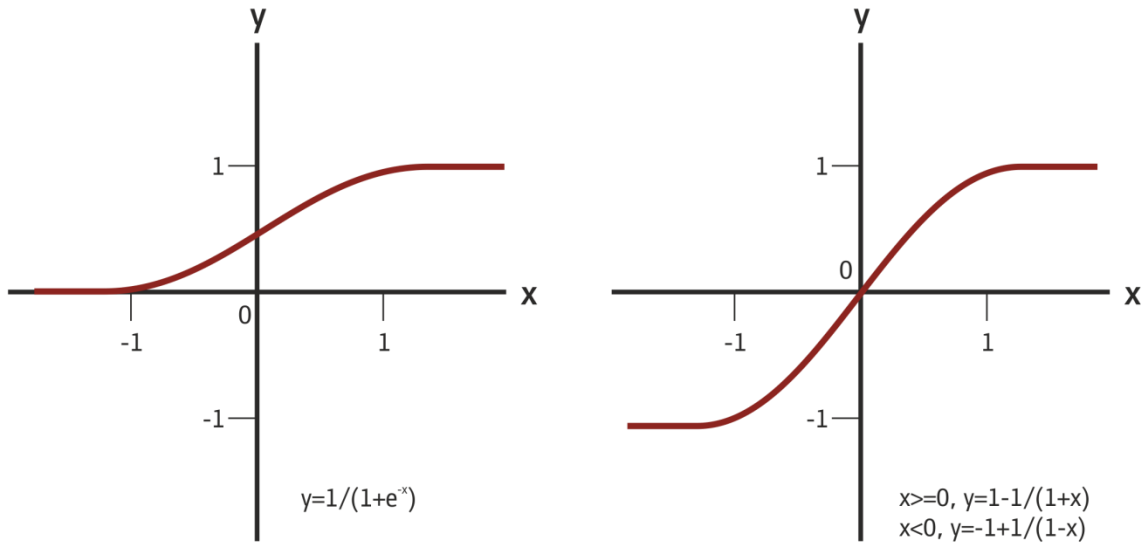


$$\begin{aligned} x < 0, & y = 0 \\ 0 \leq x \leq 1, & x = y \\ x > 1, & y = 1 \end{aligned}$$

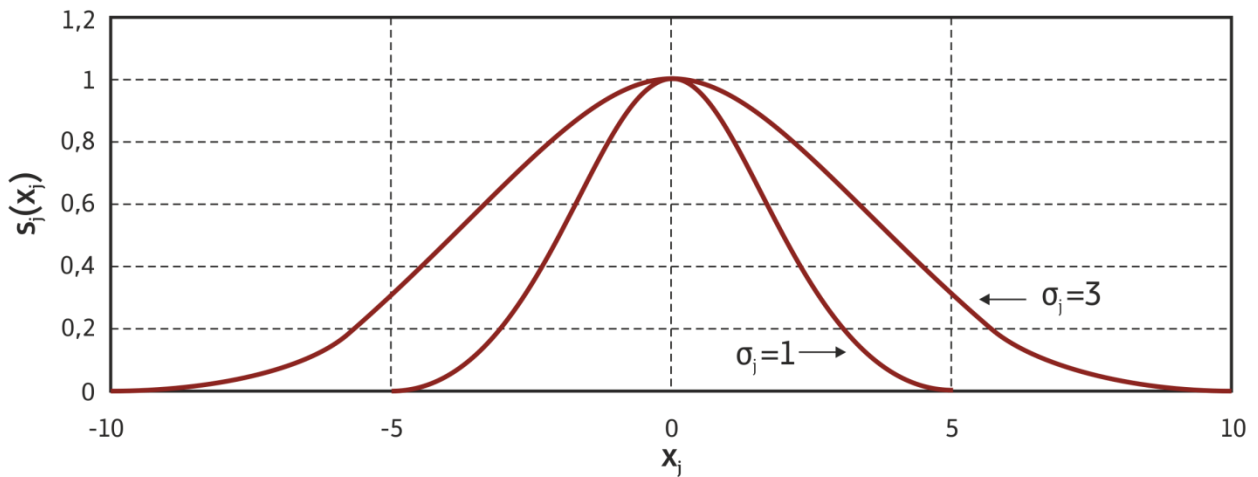
Ramping Function

Σχήμα 4.22 Γραμμικές συναρτήσεις μετάβασης

Ένα γνωστό ΤΝΔ που χρησιμοποιεί τη δυαδική γραμμική συνάρτηση είναι ο Perceptron (βλέπε [Σχήμα 4.18](#)). Συνηθέστερα, όμως, χρησιμοποιούνται μη γραμμικές συναρτήσεις, όπως οι **σιγμοειδείς συναρτήσεις** (sigmoid functions) και οι **Γκαουσιανές συναρτήσεις** (Gaussian functions) (βλέπε Σχήμα 4.23).



Σιγμοειδείς Συναρτήσεις



Γκαουσιανές Συναρτήσεις

Σχήμα 4.23 Σιγμοειδείς και Γκαουσιανές συναρτήσεις μετάβασης

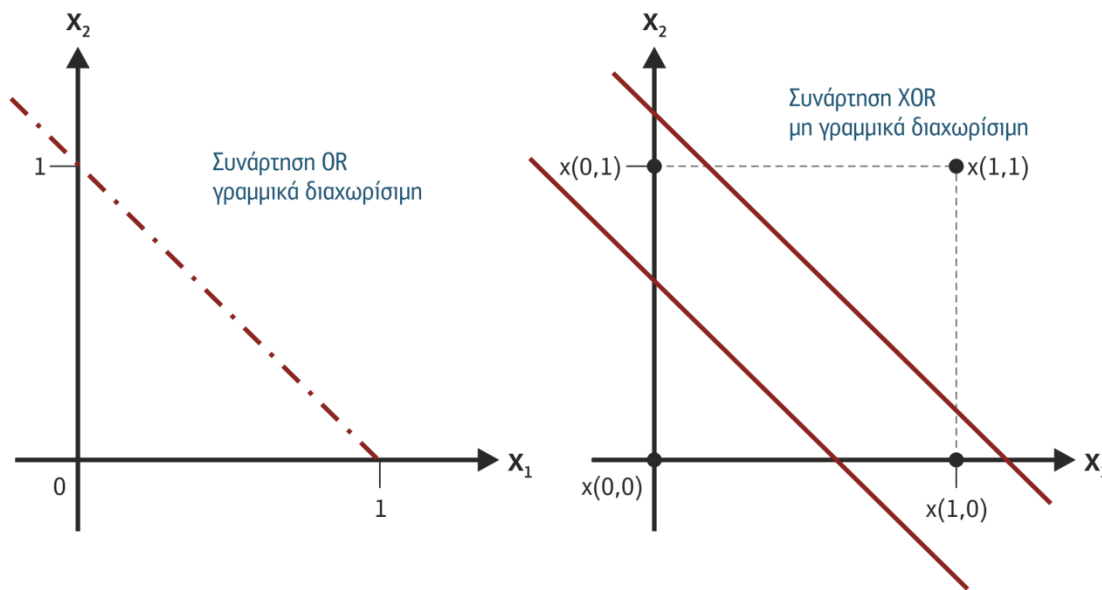
Στη Γκαουσιανή συνάρτηση που παρουσιάζεται στο σχήμα 4.23, σ_j είναι ο συντελεστής Γκαουσιανής διασποράς και c_j είναι το κέντρο που στην περίπτωση του σχήματος 4.23 είναι το 0, το δε αποτέλεσμα της υπολογίζεται ως εξής:

$$S_j(x_j) = \exp\left(-\frac{(x_j - c_j)^2}{2\sigma_j^2}\right)$$

4.5.5 ΤΝΔ Πρόσθιας Τροφοδότησης

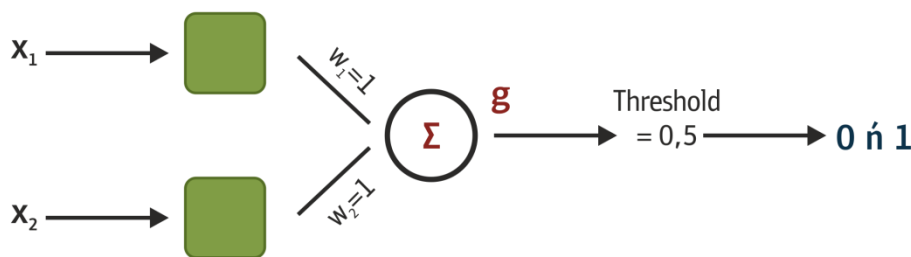
Perceptrons

Αν και ο όρος **Perceptron** (Rosenblatt,1962) χρησιμοποιήθηκε αρχικά για το στοιχειώδες ΤΝΔ πολλών εισόδων και μόνο μιας εξόδου, όπως αυτό του [σχήματος 4.18](#), έχει επικρατήσει να χαρακτηρίζονται ως Perceptrons όλα τα ΤΝΔ πρόσθιας τροφοδότησης που δεν περιέχουν στην αρχιτεκτονική τους κρυφά επίπεδα. Αν διαθέτουν την κατάλληλη δομή, οι Perceptrons είναι ικανοί να επιλύουν οποιαδήποτε γραμμική συνάρτηση, όπως την κλασική συνάρτηση με διαχωρίσιμες τιμές 'οr' (βλέπε Σχήμα 4.24 αριστερά).



Σχήμα 4.24 Διαχωρισμός τιμών συναρτήσεων

Στο σχήμα 4.25 παρουσιάζεται ένας Perceptron που επιλύει την or περιέχει. Ο Perceptron περιέχει στη δομή του δύο κόμβους εισόδου και διαθέτει έναν κόμβο εξόδου. Ως εισοδοί γίνονται δεκτές οι τιμές 0 και 1 και ως έξοδοι προκύπτει η τιμή 0 ή 1 σύμφωνα με τον πίνακα αληθείας.



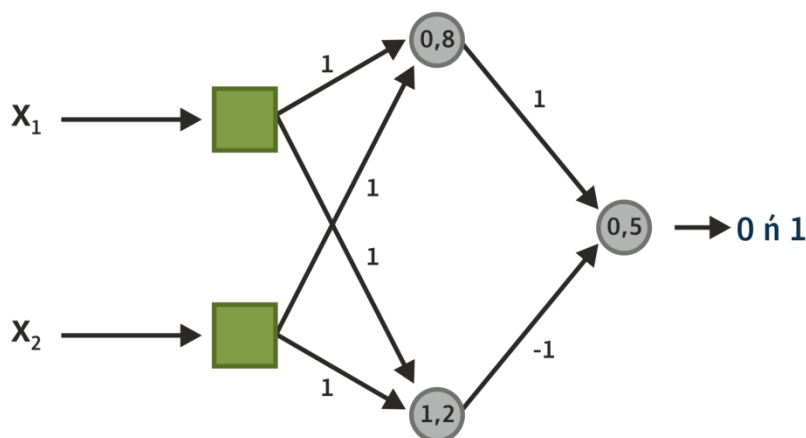
Σχήμα 4.25 Ο Perceptron που επιλύει τη γραμμικά διαχωρίσιμη συνάρτηση or

Δείτε [κινούμενη εικόνα 4.3 - Λειτουργία ΤΝΔ για την or](#)

Το ότι οι Perceptrons δεν μπορούν να επιλύσουν προβλήματα με μη γραμμικά διαχωρίσιμες τιμές εξόδου, όπως η περίπτωση του xor (βλέπε Σχήμα 4.24 δεξιά), οδήγησε στην ανάγκη προσθήκης κρυφών επιπέδων στη δομή των Perceptrons και στη δημιουργία πολυεπίπεδων ΤΝΔ.

Παράδειγμα λειτουργίας πολυεπίπεδου ΤΝΔ για το xor

Κάποια από τα απλούστερα πολυεπίπεδα ΤΝΔ ενός κρυφού επιπέδου με πρόσθια τροφοδότηση και πλήρως συνδεδεμένους κόμβους είναι αυτά που προσομοιώνουν τη λειτουργία της μη γραμμικά διαχωρίσιμης συνάρτησης xor (βλέπε [Σχήμα 4.24](#)), όπως αυτό του σχήματος 4.26, που έχει προκύψει μετά από εκπαίδευση ενός ΤΝΔ πρόσθιας τροφοδότησης με πλήρως συνδεδεμένους κόμβους. Μέσα στους κόμβους φαίνονται οι τιμές καταφλίου, ενώ στις ακμές τα βάρη. Ως συνάρτηση μετάβασης εφαρμόζεται η απλή βηματική. Ως είσοδοι γίνονται δεκτές οι τιμές 0 και 1 και η τιμή εξόδου είναι 0 ή 1 σύμφωνα με τον πίνακα αληθείας.



Σχήμα 4.26 Πολυεπίπεδο ΤΝΔ για το xor

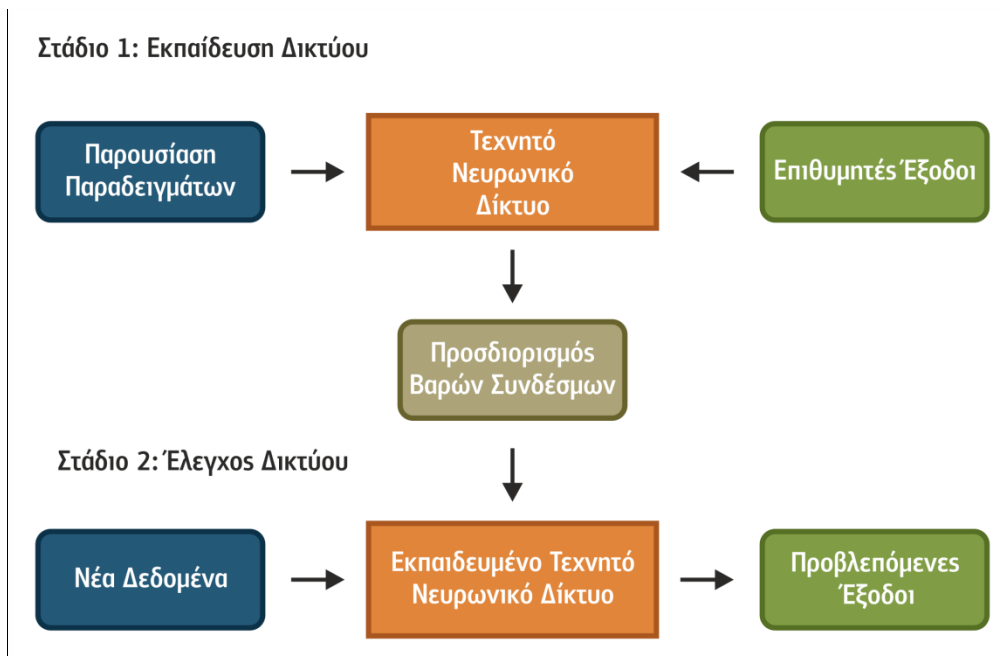
Δείτε [κινούμενη εικόνα 4.4 - Λειτουργία ΤΝΔ για την xor](#)

Στάδια ολοκλήρωσης ΤΝΔ

Αφού καθοριστεί η αρχιτεκτονική ενός ΤΝΔ, όσον αφορά τον τύπο τροφοδότησης και την εσωτερική του δομή, ακολουθούν δυο στάδια κρίσιμα για την ολοκλήρωσή του: εκπαίδευση (training) και ανάκληση (recall) (βλέπε Σχήμα 4.27).

Κατά το στάδιο της εκπαίδευσης, τα συνδεδεμένα βάρη του δικτύου “μαθαίνουν” προσαρμόζοντας τα βάρη τους.

Μετά την εκπαίδευση, ακολουθεί η φάση της ανάκλησης (δηλαδή του ελέγχου του δικτύου) με τη βοήθεια ενός νέου συνόλου δειγμάτων τα οποία δεν είχαν πάρει μέρος στη διαδικασία εκπαίδευσης.



Σχήμα 4.27 Στάδια ολοκλήρωσης ΤΝΔ

4.5.6 Εκπαίδευση ΤΝΔ

Ακριβώς για τη δυνατότητα που έχουν να μαθαίνουν μετά από εκπαίδευση, τα ΤΝΔ είναι τόσο δημοφιλή μέσα στο χώρο των συστημάτων Μηχανικής Μάθησης. Η εκπαίδευσή τους γίνεται βάσει αλγορίθμων που μπορεί να υιοθετούν μοντέλα μάθησης με επίβλεψη, χωρίς επίβλεψη ή ενισχυτικής μάθησης (βλέπε [παράγραφο 4.2](#)).

- **Εκπαίδευση με επίβλεψη**, όπου το νευρωνικό δίκτυο μαθαίνει να απεικονίζει δεδομένες εισόδους σε εξόδους εκ των προτέρων γνωστές (σύνολο εκπαίδευσης), με απώτερο στόχο τη γενίκευση της αναγνώρισης αυτής και για παρεμφερείς εισόδους στο μέλλον.
- **Εκπαίδευση χωρίς επίβλεψη**, όπου το νευρωνικό δίκτυο κατασκευάζει απεικονίσεις από μια αναπαράσταση σε μια άλλη.
- **Ενισχυτική Εκπαίδευση**, όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών για δεδομένου τύπου παρατηρήσεις.

Η εκπαίδευση έχει ως βασικό στόχο να βρεθεί ένας τρόπος αλλαγής των συνδεσμικών βαρών που θα έχει ως αποτέλεσμα την αλλαγή της γενικής συμπεριφοράς του δικτύου με την αύξηση της ικανότητας του δικτύου να παρέχει στο μέλλον μία επιθυμητή έξοδο μετά από μία δεδομένη είσοδο. Όταν η επιθυμητή έξοδος είναι εκ των προτέρων γνωστή λέμε ότι το δίκτυο μαθαίνει με επίβλεψη (supervised learning), αλλιώς μαθαίνει χωρίς επίβλεψη (unsupervised learning). Τα ΤΝΔ με οπισθοδρόμηση ανήκουν στην πρώτη περίπτωση, ενώ στη δεύτερη ανήκει το δίκτυο Kohonen. Ο πίνακας 4.4 παρουσιάζει τα βασικότερα είδη δικτύων και τον τρόπο με τον οποίο μαθαίνουν.

Γνωστότερα Είδη ΤΝΔ			
Όνομα ΤΝΔ	Κατασκευαστής	Έτος	Τρόπος Εκπαίδευσης
Perception	Rosenblatt (USA)	1957-62	Με επίβλεψη
Adaline/Madaline	Widrow (USA)	1960-62	Με επίβλεψη
Back-propagation	Werbos, Rumelhart, et al.	1974-86	Με επίβλεψη
Self-organising map	Kohonen (Finland)	1981	Χωρίς επίβλεψη
Hopfield Net	Hopfield (USA)	1982	Με επίβλεψη
Boltzmann machine	Hinton (Canada), Hopkins, Szu (USA)	1985-86	Με επίβλεψη

Πίνακας 4.4 Γνωστότερα είδη ΤΝΔ

Βασικό στοιχείο της αρχιτεκτονικής ενός ΤΝΔ είναι ο τρόπος ελέγχου της αλλαγής των βαρών κατά την εκπαίδευση, δηλαδή ο **αλγόριθμος εκπαίδευσης** (training algorithm) που υλοποιείται αποκλειστικά από το ίδιο το δίκτυο χωρίς εξωτερική επέμβαση. Γνωστότεροι αλγόριθμοι εκπαίδευσης είναι οι ακόλουθοι:

- Αλγόριθμος οπισθοδιάδοσης λάθους (backpropagation),
- Ανταγωνιστική μάθηση (competitive learning),
- Τυχαία μάθηση (random learning).

Κάθε αλγόριθμος εκπαίδευσης χρησιμοποιεί κάποιον κανόνα εκμάθησης, για να προσαρμόσει τα συνδεσμικά βάρη μεταξύ των νευρώνων του. Ο απλούστερος αλγόριθμος επιβλεπόμενης εκπαίδευσης είναι αυτός του στοιχειώδους perceptron, ο οποίος μπορεί να εφαρμοστεί και σε οποιονδήποτε απλό νευρώνα ενός ΤΝΔ.

Εκπαίδευση Perceptron

Η βασική ιδέα της εκπαίδευσης ενός Perceptron που γίνεται πάντα κάτω από επίβλεψη είναι να γίνει χρήση ενός συνόλου εκπαίδευσης (training set), όπου για κάθε ζεύγος $(x, f(x))$ ο αλγόριθμος εκπαίδευσης πραγματοποιεί τα ακόλουθα βήματα:

- Ελέγχει εάν η y που δίνει ο νευρώνας για το δείγμα x είναι η αναμενόμενη.
- Εάν είναι, η διαδικασία εκπαίδευσης προχωρά στο επόμενο δείγμα.
- Εάν όχι, τότε:
 - εάν η σωστή έξοδος είναι μεγαλύτερη από αυτήν που υπολόγισε ο νευρώνας, ο κανόνας εκμάθησης αυξάνει τα βάρη των εισόδων που είναι θετικές και μειώνει τα βάρη των εισόδων που είναι αρνητικές.
 - εάν η σωστή έξοδος είναι μικρότερη από αυτήν που υπολόγισε ο νευρώνας, ο κανόνας εκμάθησης μειώνει τα βάρη των εισόδων που είναι θετικές και αυξάνει τα βάρη των εισόδων που είναι αρνητικές.
- Η διαδικασία αυτή εκτελείται, μέχρις ότου ο νευρώνας να απαντά σωστά σε όλα τα δείγματα ή να μη βελτιώνει πλέον σημαντικά την απόδοσή του.

Αλγόριθμος Εκπαίδευσης Perceptron με δυο κλάσεις ταξινόμησης, A και B

Βήμα1: Αρχικοποίηση

Θέσε $w(0)=0$.

Βήμα 2: Ενεργοποίηση

Στο χρόνο n , ενεργοποίησε τον Perceptron εφαρμόζοντας το συνεχές διάνυσμα εισόδου $x(n)$ και το $d(n)$.

Βήμα 3: Υπολογισμός πραγματικής απόκρισης

Υπολόγισε την πραγματική απόκριση του Perceptron:

$$y(n) = \text{sgn}[w^T(n) x(n)]$$

Βήμα 4: Προσαρμογή διανύσματος βαρών

Προσάρμοσε τα βάρη του Perceptron.

$$w(n+1) = w(n) - n [d(n) - y(n)] x(n)$$

Όπου:

$$d(n) = \begin{cases} +1, & \text{αν το } x(n) \text{ ανήκει στην κλάση } A \\ -1, & \text{αν το } x(n) \text{ ανήκει στην κλάση } B \end{cases}$$

Βήμα 5: Αύξησε το χρόνο κατά μια μονάδα και πήγαινε στο βήμα 2.

Κάθε μονάδα μέσα σε ένα ΤΝΔ είναι ανεξάρτητη από τις υπόλοιπες και ομοίως είναι και η εκπαίδευσή της. Άρα, στα δίκτυα που μαθαίνουν υπό επίβλεψη κάθε μονάδα μπορεί να θεωρηθεί ως ένας στοιχειώδης Perceptron και να ακολουθήσει την περιγραφείσα διαδικασία εκπαίδευσης.

Κανόνες Εκμάθησης

Οι αλγόριθμοι εκπαίδευσης χαρακτηρίζονται από τους **κανόνες εκμάθησης** (learning rules) που χρησιμοποιούνται, για να υπολογίσουν τα σφάλματα και να διορθώσουν τα συνδεδεσμένα βάρη των εσωτερικών νευρώνων του δικτύου. Ένας από τους γνωστότερους κανόνες μάθησης που χρησιμοποιείται για εκπαίδευση είναι ο **κανόνας Δέλτα** (Delta rule), που χρησιμοποιεί τη βηματική συνάρτηση ενεργοποίησης και υπολογίζει τη μεταβολή βάρους ως ακολούθως:

$$w_i = w_{iold} - d * (a_j - a_i)$$

όπου:

- a_i : η τρέχουσα έξοδος του νευρώνα i ,
- a_j : η επιθυμητή έξοδος του νευρώνα i για το τρέχον δείγμα,
- d : ο ρυθμός εκμάθησης ($d>0$),
- w_i : το νέο βάρος εισόδου από τον νευρώνα i ,
- w_{iold} : το παλιό βάρος εισόδου από τον νευρώνα i .

Ο ρυθμός εκμάθησης d καθορίζει το πόσο γρήγορα συγκλίνει η μάθηση. Μεγάλος ρυθμός μάθησης μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση και σε ταλάντωση γύρω από τις βέλτιστες τιμών βαρών. Μικρός ρυθμός μάθησης έχει ως αποτέλεσμα πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσει σε παγίδευση σε τοπικά ακρότατα.

Άλλος Κανόνας Εκμάθησης είναι ο **Κανόνας του Hebb** (Hebbian rule):

$$\Delta weight_{ui} = Irate * a_u * a_i$$

όπου

- $Irate$: σταθερά που καθορίζει το ρυθμό εκμάθησης,
- a_u, a_i : τιμές ενεργοποίησης στις μονάδες u και i .

Εάν $a_u \cdot a_i > 0$ (και οι δυο + ή -), τότε οι συνδέσεις μεταξύ τους ενισχύονται αναλογικά με το αποτέλεσμα των δυο ενεργοποιήσεων. Αλλιώς, μειώνονται αναλογικά.

Σφάλματα

Ως σφάλμα μιας εξόδου o ενός νευρώνα k από μια επιθυμητή του έξοδο a , αναφορικά με ένα δείγμα p , ορίζεται η ποσότητα:

$$E_k = (a_{k,p} - o_{k,p})$$

Το συνολικό σφάλμα για όλα τα παραδείγματα εκπαίδευσης για το νευρώνα k είναι το μέσο τετραγωνικό σφάλμα:

$$E = \frac{1}{P} \sum_p (a_{k,p} - o_{k,p})^2$$

Με παρόμοιο τρόπο μπορούμε να ορίσουμε το μέσο τετραγωνικό σφάλμα για όλους τους νευρώνες εξόδου k και όλα τα παραδείγματα p :

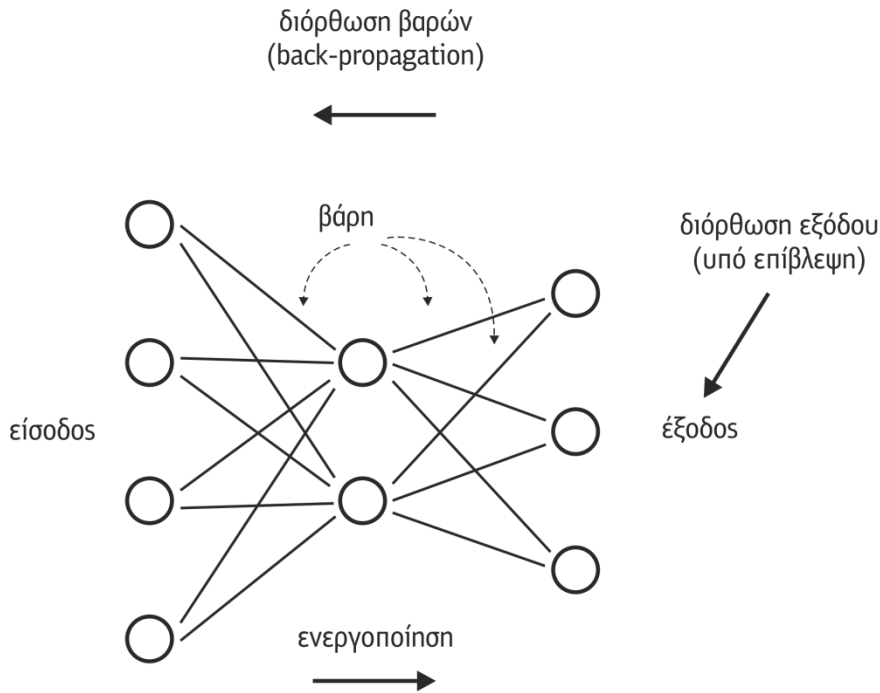
$$E = \frac{1}{P * K} \sum_{p=1}^P \sum_{k=1}^K (a_{k,p} - o_{k,p})^2$$

Ο τερματισμός της διαδικασίας εκπαίδευσης πραγματοποιείται, όταν το συνολικό σφάλμα E για όλα τα παραδείγματα και για όλους τους νευρώνες εξόδου πέσει κάτω από μια μικρή τιμή.

Εκπαίδευση ΤΝΔ με τη μέθοδο οπισθοδιάδοσης σφάλματος

Στα πολυεπίπεδα ΤΝΔ, ο συνηθέστερος τρόπος επιβλεπόμενης μάθησης είναι η **μάθηση με οπισθοδιάδοση σφάλματος** (error back-propagation). Τα πολυεπίπεδα ΤΝΔ στα οποία ακολουθείται αυτή η μέθοδος εκπαίδευσης, καλούνται **Backpropagation ΤΝΔ** (Backpropagation ANN).

Κατά την εκπαίδευση των Backpropagation ΤΝΔ, για κάθε είσοδο που δίνεται στο δίκτυο υπολογίζονται οι εξοδοί με εφαρμογή των συναρτήσεων μετάβασης σε κάθε μονάδα κρυφού ή εξωτερικού επιπέδου (βλέπε Σχήμα 4.28). Για κάθε μονάδα εξωτερικού επιπέδου λαμβάνονται υπόψη οι διαφορές μεταξύ του υπολογιζόμενου και του επιθυμητού αποτελέσματος και διαδίδονται προς τα πίσω στις μονάδες του κρυφών επιπέδων, έτσι ώστε να καθορίσουν τις απαραίτητες αλλαγές στα βάρη σύνδεσης μεταξύ των μονάδων. Οι αλλαγές αυτές γίνονται βάσει του κανόνα εκμάθησης οπισθοδιάδοσης και ως σκοπό έχουν τη μείωση του εμφανιζόμενου στην έξοδο σφάλματος.



Σχήμα 4.28 Εκπαίδευση δικτύου με οπισθοδιάδοση

Η συμμετοχή μιας μονάδας στα σφάλματα των μονάδων του επόμενου επιπέδου της είναι ανάλογη της τρέχουσας εισόδου της και των συντελεστών βαρύτητας που τη συνδέουν με τις μονάδες του επόμενου επιπέδου. Το δίκτυο εν συνεχεία εφαρμόζει εκ νέου τις συναρτήσεις μετάβασης, για να υπολογίσει το νέο σφάλμα.

Το πραγματικό σφάλμα E_k μιας μονάδας εξόδου k ενός παραδείγματος p υπολογίζεται καταρχάς όπως και στον Perceptron, δηλαδή:

$$E_k = (a_{kp} - o_{kp})$$

Στη συνέχεια πολλαπλασιάζεται επί την παράγωγο της συνάρτησης ενεργοποίησης στη μονάδα k (u_k), σύμφωνα με τον γενικευμένο κανόνα δέλτα, για να υπολογιστεί το λεγόμενο **προσαρμοσμένο σφάλμα νευρώνα**:

$$\delta_k = (a_{kp} - o_{kp}) \cdot g'(u_k)$$

Το αντίστοιχο σφάλμα σε μια μονάδα κρυφού επιπέδου i υπολογίζεται από τα προσαρμοσμένα σφάλματα στις k μονάδες του επόμενου επιπέδου με τις οποίες η μονάδα συνδέεται με βάρη w_{ik} , ως εξής:

$$\delta_i = g'(u_i) \cdot \sum_1^k w_{ik} \cdot \delta_k$$

Η διαδικασία της εκπαίδευσης περιλαμβάνει πλήθος τέτοιων κύκλων διόρθωσης σφάλματος και λήγει με τη μείωση του σφάλματος κάτω από ένα επιθυμητό όριο. Εναλλακτικά, ως συνθήκη τερματισμού μπορεί να θεωρηθεί η πραγματοποίηση ενός συγκεκριμένου αριθμού κύκλων εκπαίδευσης ή η πάροδος ενός συγκεκριμένου χρονικού διαστήματος.

Αλλαγές στα βάρη

Αφού έχει υπολογισθεί για κάθε μονάδα i το σφάλμα δ_i , η αλλαγή στα βάρη εισόδου σε όλους τους νευρώνες γίνεται ως εξής:

$$\Delta w_{ji} = -d \cdot \delta_i \cdot a_j$$

Δηλαδή, η αλλαγή στο βάρος από τον νευρώνα i στον νευρώνα j εξαρτάται από το σφάλμα του νευρώνα i , την έξοδο του νευρώνα j και το ρυθμό μάθησης (learning rate) d .

Υπάρχουν 2 τρόποι αλλαγής των βαρών κατά την παρουσίαση των παραδειγμάτων εκπαίδευσης:

- **Αυξητική εκπαίδευση** (incremental training), όπου για κάθε δείγμα που παρουσιάζεται γίνονται άμεσα οι αλλαγές των βαρών
- **Μαζική εκπαίδευση** (batch training), όπου πρώτα παρουσιάζονται όλα τα παραδείγματα μια φορά, υπολογίζονται μόνο οι αλλαγές των βαρών για κάθε δείγμα και εφαρμόζονται ταυτόχρονα αφού παρουσιαστούν όλα τα παραδείγματα.

Η παρουσίαση όλων των παραδειγμάτων μια φορά (ανεξαρτήτως του τρόπου αλλαγής των βαρών) αντιστοιχεί σε έναν κύκλο εκπαίδευσης και ονομάζεται **εποχή εκπαίδευσης** (training epoch).

Μη Επιβλεπόμενη Μάθηση ΤΝΔ – Δίκτυα Kohonen

Αυτό το μοντέλο των δικτύων προτάθηκε το 1984 από τον Kohonen και αφορά δίκτυα που ακολουθούν διαδικασία εκπαίδευσης χωρίς επίβλεψη δεν υπάρχει καμία εξωτερική επέμβαση ως προς το τι πρέπει να εκπαιδευθεί ένα δίκτυο να αναγνωρίζει. Το χαρακτηριστικό του προτύπου αυτού είναι ότι μπορεί να ταξινομεί διανύσματα με τη βοήθεια ενός αλγόριθμου μη επιβλεπόμενης μάθησης.

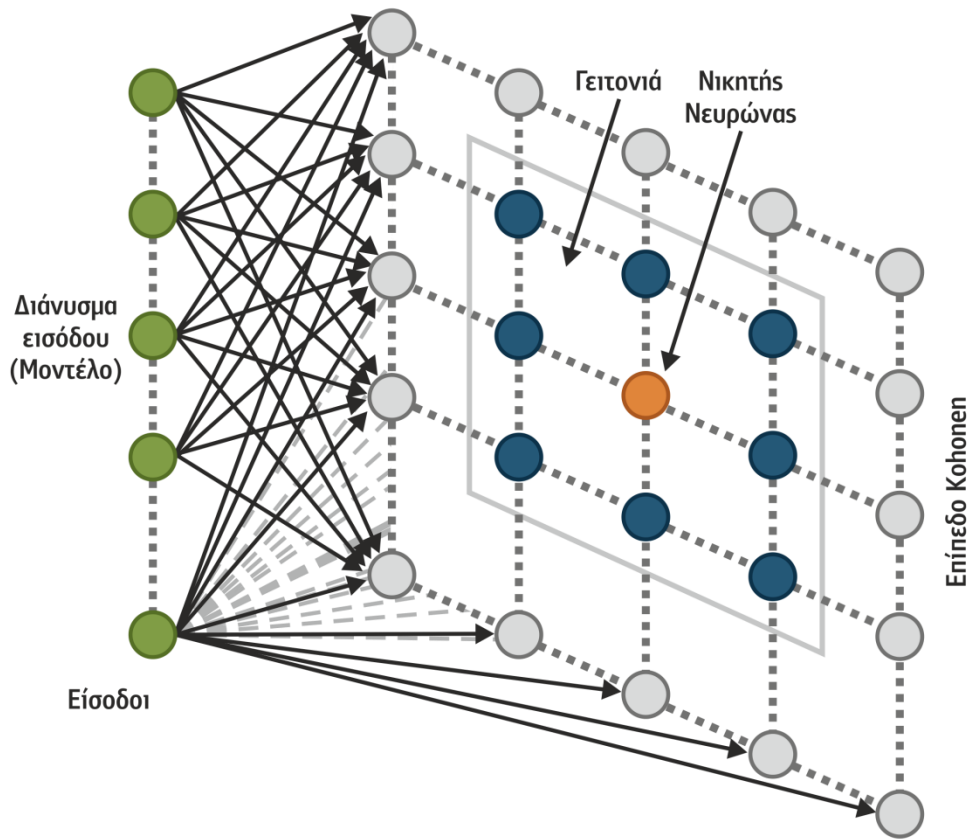
Το δίκτυο Kohonen οργανώνει τον πίνακα των βαρών του με τέτοιο τρόπο, ώστε αναγνωρίζει όποια κανονικότητα υπάρχει στα διανύσματα εισόδου. Στο σημείο αυτό προσομοιώνει μία σημαντική αρχή της οργάνωσης των αισθητηρίων οργάνων του εγκεφάλου: η κατανομή των νευρώνων παρουσιάζει κανονικότητα που αντικατοπτρίζει κάποια ειδικά χαρακτηριστικά των εξωτερικών ερεθισμάτων που διαδίδονται σε αυτά.

Κάθε δίκτυο Kohonen αποτελείται από δύο επίπεδα: το 1ο είναι το επίπεδο εισόδου και το 2ο καλείται **επίπεδο Kohonen** (Kohonen layer), με ιδιαίτερο χαρακτηριστικό ότι είναι οργανωμένο σε μορφή πλέγματος, το οποίο μπορεί να έχει οποιαδήποτε διάσταση: για παράδειγμα, μπορεί να έχουμε ένα διδιάστατο πλέγμα, δηλαδή μία επιφάνεια που έχει επάνω της $n \times m$ μονάδες που αντιστοιχούν στους νευρώνες. Τα δύο αυτά επίπεδα έχουν πλήρη συνδεσμολογία, δηλαδή κάθε μονάδα εισόδου συνδέεται με όλες τις μονάδες του επιπέδου Kohonen. Αν το επίπεδο εισόδου αντιστοιχεί σε ένα διάνυσμα με k στοιχεία εισόδου, τελικά θα έχουμε $k \times n \times m$ συνδέσεις. Το σχήμα 4.29 δείχνει την τυπική δομή ενός τέτοιου δικτύου.

Η εκπαίδευση ενός δικτύου Kohonen περιλαμβάνει την παρουσίαση στο δίκτυο ενός μοντέλου (pattern) και τελικώς την εύρεση στο επίπεδο Kohonen εκείνων των μονάδων για τις οποίες ισχύει ότι το άθροισμα των τιμών ενεργοποίησης που καταλήγουν σε αυτές έχουν τις υψηλότερες τιμές (δηλαδή μια μονάδα ξεχωρίζει από τις υπόλοιπες σε σχέση με τη συνολική ενέργεια που δέχεται). Τα βάρη της μονάδας που “κέρδισε” και όλων των μονάδων που αποτελούν τη *γειτονιά ή συστάδα* (cluster) ρυθμίζονται βάσει του νόμου εκμάθησης. Στην αρχή της περιόδου εκπαίδευσης, η γειτονιά είναι ένα μεγάλο μέρος του επιπέδου Kohonen, μέσα σε μια προκαθορισμένη ακτίνα από τη μονάδα που κερδίζει. Όμως, η ακτίνα αυτή μειώνεται προοδευτικά όσο προχωράει η εκπαίδευση. Η προσαρμογή των βαρών γίνεται με τέτοιο τρόπο, ώστε μετά την εκπαίδευση το δίκτυο να είναι σε θέση να διεγείρει την ίδια πάντα μονάδα εξόδου για διανύσματα εισόδου που ανήκουν στην ίδια κατηγορία.

Επειδή η εκπαίδευση γίνεται αυτόνομα (χωρίς στόχους), δεν μπορούμε να γνωρίζουμε εκ των προτέρων σε ποια από τις τάξεις θα αντιστοιχεί η καθεμία από τις μονάδες εξόδου, με αποτέλεσμα η αντιστοίχιση να γίνεται μόνο μετά την εκπαίδευσή τους με το σχηματισμό γειτονιών συσχετισμένων με τις επιδιωκόμενες τάξεις. Ο κανόνας εκμάθησης μπορεί να διαφέρει από μονάδα σε μονάδα της γειτονιάς, αλλά συνήθως ακολουθεί τη γνωστή συνάρτηση του μεξικάνικου καπέλου (mexican hat function).

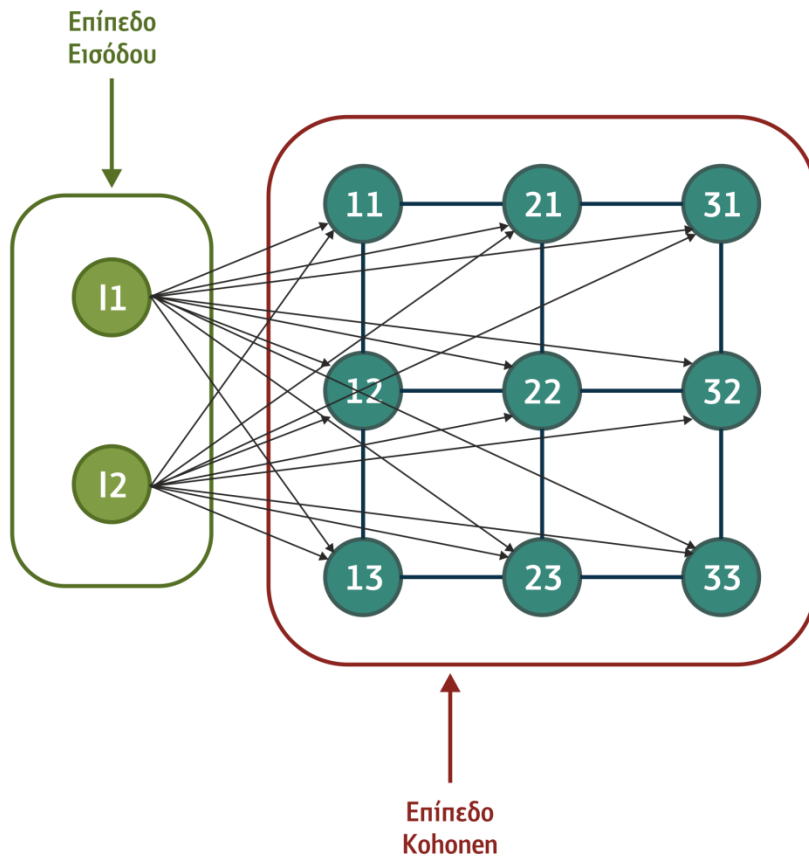
Λόγω του τρόπου εκπαίδευσής τους, τα δίκτυα Kohonen καλούνται και **αυτο-οργανωνόμενοι χάρτες** (self-organizing maps).



Σχήμα 4.29 Τυπικό δίκτυο Kohonen

Παράδειγμα εκπαίδευσης δικτύου Kohonen

Έστω το δίκτυο Kohonen του σχήματος 4.30 το οποίο θέλουμε να εκπαιδευτεί να αναγνωρίζει δυο τάξεις κατηγοριοποίησης, δηλαδή να “μάθει” 2 γειτονιές. Το επίπεδο εισόδου του δικτύου αποτελείται από 2 μονάδες, ενώ το επίπεδο Kohonen διαθέτει 9 νευρώνες.

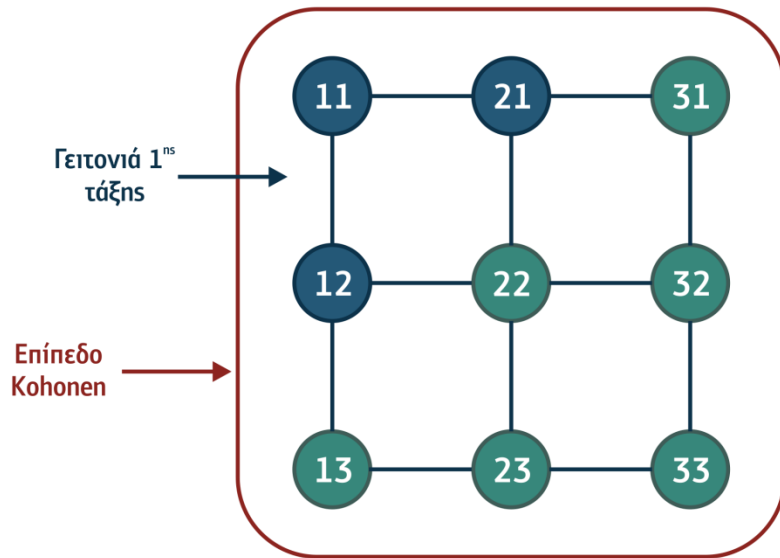


Σχήμα 4.30 Παράδειγμα δικτύου Kohonen

Η εκπαίδευση θα πραγματοποιηθεί σε 2 κύκλους εκπαίδευσης.

- 1^{ος} : το δίκτυο θα μάθει να αναγνωρίζει την 1η τάξη διαχωρισμού
- 2^{ος} : η εκπαίδευση θα επαναληφθεί για το ήδη εκπαιδευμένο δίκτυο, ώστε να μάθει και τη δεύτερη τάξη.

Στο παράδειγμά μας, κατά τον 1^ο κύκλο το σύνολο εκπαίδευσης θα αποτελείται από δείγματα μοντέλων της 1^{ης} τάξης διαχωρισμού. Στην αρχή, τα συνδεσμικά βάρη θα είναι τυχαία και, αφού υπολογιστεί η ενέργεια που δέχεται κάθε μονάδα του επιπέδου Kohonen, θα επιλεγεί η νικήτρια μονάδα και η γειτονιά της. Στην περίπτωσή μας η νικήτρια μονάδα είναι η 11 και η γειτονιά της αποτελεί τις μονάδες 12 και 21. Οι 7 πιο αδύνατες μονάδες του επιπέδου θα εξαιρεθούν της διαδικασίας εκπαίδευσης και οι υπόλοιπες ως μια γειτονιά θα συμμετέχουν στη διαδικασία, ώστε να διαμορφωθούν τα τελικά βάρη τους (βλέπε Σχήμα 4.31). Στον 2^ο κύκλο, θα διατηρηθούν τα βάρη της 1^{ης} γειτονιάς και ο κύκλος θα επαναληφθεί με δείγματα μοντέλων που ανήκουν στη 2^η τάξη διαχωρισμού.



Σχήμα 4.31 Γειτονιά μετά τον 1ο κύκλο εκπαίδευσης δικτύου Kohonen

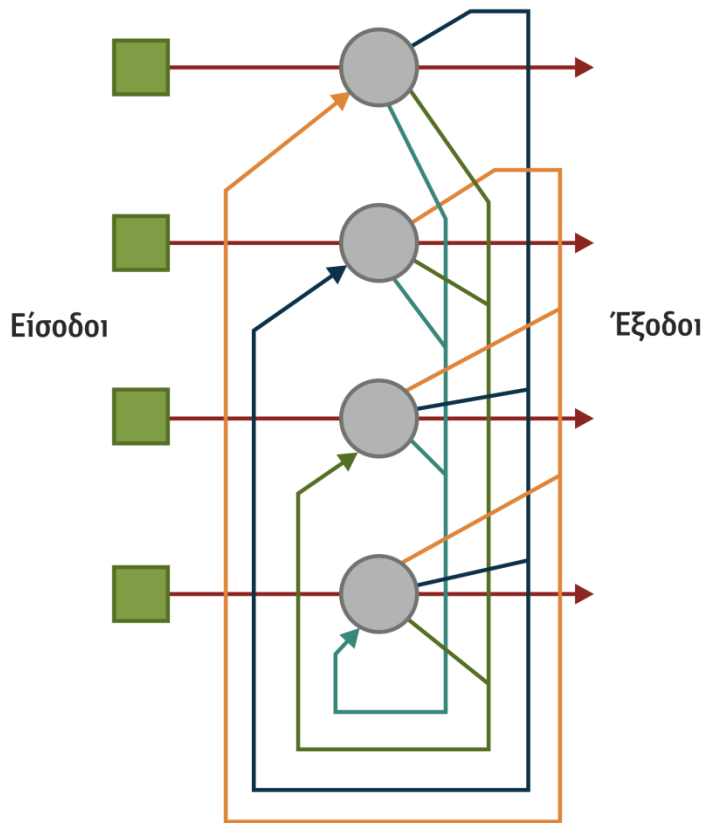
Τα δίκτυα Kohonen έχουν μεγάλη εφαρμογή στην εξόρυξη δεδομένων (data mining) κυρίως μέσα από μεγάλες βάσεις δεδομένων στο διαδίκτυο.

Ανατροφοδοτούμενα ΤΝΔ - Δίκτυα Hopfield

Ένα δίκτυο Hopfield είναι μια μη-γραμμική **συσχετιστική μνήμη** (autoassociative memory) ή **μνήμη διευθυνσιοδοτούμενη από τα περιεχόμενα**. Η κύρια λειτουργία μιας τέτοιας μνήμης είναι η ανάκτηση ενός προτύπου, που έχει αποθηκευθεί σε αυτήν. Τα δίκτυα Hopfield ανήκουν στην κατηγορία των ανατροφοδοτούμενων (recurrent) ΤΝΔ.

Η δομή των δικτύων Hopfield έχει τα ακόλουθα χαρακτηριστικά:

- Υπάρχει μόνο ένα επίπεδο, αλλά με πολλές μονάδες (νευρώνες).
- Είναι δυαδικά συστήματα: κάθε μονάδα χαρακτηρίζεται από μία δυαδική κατάσταση, δηλαδή μπορεί να έχει μία από δύο δυνατές τιμές. Συνήθως οι τιμές αυτές είναι 0 και 1, αλλά αυτό δεν είναι απαραίτητο.
- Οι συνδέσεις μεταξύ των νευρώνων είναι πλήρεις, δηλαδή κάθε μονάδα συνδέεται με κάθε άλλη μονάδα στο δίκτυο.
- Όλοι οι νευρώνες λειτουργούν ταυτόχρονα ως είσοδοι και έξοδοι του δικτύου.



Σχήμα 4.32 Αρχιτεκτονική του Δικτύου Hopfield

Σε ένα δίκτυο Hopfield με n μονάδες και δομή όπως περιγράφηκε παραπάνω, θα έχουμε $n \times (n-1)$ συνδέσεις, διότι κάθε μονάδα συνδέεται με κάθε άλλη μονάδα, αλλά όχι με τον εαυτό της (βλέπε Σχήμα 4.32). Στη γενική περίπτωση, οι συνδέσεις έχουν και συγκεκριμένη κατεύθυνση. Έτσι, σε κάθε ζευγάρι μονάδων που συνδέονται υπάρχει σύνδεση και προς τις δύο κατευθύνσεις, δηλαδή μεταξύ των μονάδων i και j υπάρχουν η σύνδεση w_{ij} και η σύνδεση w_{ji} . Γενικά, στα δίκτυα Hopfield θα ισχύει πάντοτε ότι $w_{ij}=w_{ji}$, διότι τότε εξασφαλίζεται ότι το δίκτυο συγκλίνει και καταλήγει σε μία σταθερή κατάσταση.

Η μάθηση στα δίκτυα Hopfield είναι επιβλεπόμενη με συνάρτηση μετάβασης τη συνάρτηση προσήμου. Χαρακτηριστικό των δικτύων αυτών είναι ότι τα βάρη του δικτύου συνεχώς αναπροσαρμόζονται κατά τη φάση της εκπαίδευσης. Κατά την εκπαίδευση, η κατάσταση των μονάδων αλλάζει (από 0 σε 1 και αντίστροφα) ή παραμένει η ίδια. Η εκπαίδευση ολοκληρώνεται όταν δεν πραγματοποιούνται πλέον άλλες αλλαγές.

Τα δίκτυα Hopfield χρησιμοποιούνται σε προβλήματα βελτιστοποίησης.

4.5.7 Προτερήματα ΤΝΔ

Τα ΤΝΔ διαφέρουν αισθητά από άλλες τεχνικές υλοποίησης υπολογιστικών συστημάτων, που επιλύουν προβλήματα παρόμοια με αυτά που επιλύουν στα ΤΝΔ, στα παρακάτω:

- Δεν χρησιμοποιούν σύμβολα για αναπαράσταση εννοιών του μοντέλου.
- Δεν κάνουν (σαφή) προγραμματισμό της συμπεριφοράς του μοντέλου.
- Αυτοπρογραμματίζονται “μαθαίνοντας” να παράγουν συγκεκριμένες εξόδους, όταν δίνονται κάποιες είσοδοι.

Οι παραπάνω διαφορές χαρακτηρίζουν και τα προτερήματά τους που παραθέτονται στη συνέχεια:

Όσον αφορά το Σχεδιασμό:

- Ένα ΤΝΔ σχεδιάζεται σε αναλογία με τον ανθρώπινο εγκέφαλο. Οι μηχανικοί αντλούν από τη νευροβιολογία νέες ιδέες για την επίλυση πολύπλοκων προβλημάτων.
- Επίσης σχεδιάζεται για να παρέχει πληροφορίες όχι μόνο για το συγκεκριμένο σύνολο υποδειγμάτων με το οποίο εκπαιδεύεται αλλά και για όποιο καινούργιο υπόδειγμα παρουσιαστεί.
- Τα ΤΝΔ παρουσιάζουν ομοιομορφία ανάλυσης και σχεδιασμού, με την έννοια ότι ο ίδιος συμβολισμός χρησιμοποιείται σε όλα τα επιστημονικά πεδία που περιέχουν εφαρμογές των ΤΝΔ, επιτρέποντας λόγω αυτού τη διάχυση της σχετικής τεχνογνωσίας.

Όσον αφορά την Υλοποίηση:

- Ένα ΤΝΔ δομείται από τη σύνδεση νευρώνων, οι οποίοι θεωρούνται μη-γραμμικές συσκευές. Η μη-γραμμικότητα είναι πολύ σημαντική ιδιότητα, αν ο φυσικός μηχανισμός για την παραγωγή των σημάτων εισόδου είναι μη-γραμμικός.
- Η γνώση, κατά την είσοδο στο ΤΝΔ, αναπαρίσταται από μια διασκορπισμένη μορφή (ένα σύνολο από τιμές χαρακτηριστικών σε μορφή συμβολοσειράς) και όχι από μια τεχνικά κατασκευασμένη συμβολική μορφή (π.χ. δομημένη εγγραφή ενός αρχείου ή μιας βάσης δεδομένων).

Όσον αφορά τη Δομή:

- Τα ΤΝΔ έχουν τη δυνατότητα να προσαρμόζουν τα βάρη τους στις αλλαγές του περιβάλλοντός τους επαναλαμβάνοντας απλώς την εκπαίδευσή τους με νέο σύνολο παραδειγμάτων από το μεταβαλλόμενο περιβάλλον.
- Ένα ΤΝΔ υλοποιημένο σε υλικό (hardware) έχει τη σημαντική ιδιότητα να είναι “ανεκτικό σε σφάλματα”. Για παράδειγμα, αν μια σύνδεση ενδιάμεσου επιπέδου διαγραφεί ή ένα βάρος αλλοιωθεί, η λειτουργία το δικτύου δε θα επηρεαστεί γιατί το συνολικό μέσο σφάλμα δε θα αλλάξει σημαντικά.
- Η δομή των ΤΝΔ σε επίπεδα τους δίνει δυνατότητες παράλληλων λειτουργιών και κάνει δυνατή την υλοποίησή τους σε VLSI (Very-Large-Scale Integration) τεχνολογία που τα καθιστά κατάλληλα, μεταξύ άλλων τεχνολογιών, να συμπεριληφθούν σε εφαρμογές πραγματικού χρόνου.

4.5.8 ΤΝΔ και Ανθρώπινος Εγκέφαλος

Υπάρχουν πολλά χαρακτηριστικά των ΤΝΔ κοινά με συμπεριφορές του ανθρώπινου εγκεφάλου. Πιο συγκεκριμένα μπορούν να παρατηρηθούν τα παρακάτω:

1. Η συμπεριφορά τους είναι βιολογικά ερμηνεύσιμη.
2. Δίνουν συγκεκριμένες απαντήσεις σε συγκεκριμένα ερεθίσματα.
3. Μαθαίνουν με επαναληπτική παρουσίαση παραδειγμάτων.
4. Λειτουργούν ικανοποιητικά ακόμα και μετά από καταστροφή ορισμένων κόμβων ή με ελλείψεις εισόδου.
5. Αναγνωρίζουν και κατηγοριοποιούν νέες πληροφορίες παρόμοιες με ήδη κατηγοριοποιημένες.
6. Εξηγούν το φαινόμενο της αμνησίας λόγω του ότι η ορθή λειτουργία ενός εκπαιδευμένου δικτύου μπορεί να αλλοιωθεί αν καταστραφεί ένα σημαντικό μέρος της δομής του αλλά μόλις αποκατασταθεί η λειτουργία του θα επανέλθει κανονικά δεδομένου ότι τα βάρη που έχουν

δημιουργηθεί μεταξύ των μονάδων κατά την εκπαίδευση εξακολουθούν να υφίστανται αναλλοίωτα.

4.5.9 Εφαρμογές ΤΝΔ

Τα ΤΝΔ είναι δημοφιλή σε προβλήματα τα οποία περιέχουν μη-προβλέψιμες λειτουργίες και δεν είναι πλήρως κατανοητά (κατηγοριοποίηση, αποτίμηση, πρόβλεψη):

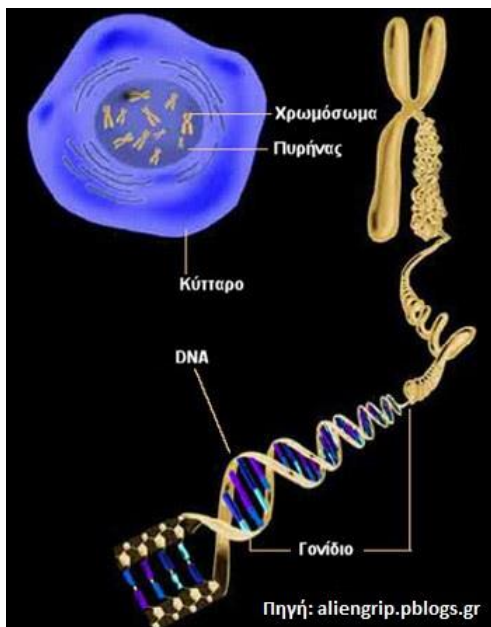
- αναγνώριση εικόνας,
- επεξεργασία φωνής,
- σχεδιασμός ενεργειών (planning),
- εξόρυξη πληροφορίας,
- χρονοπρογραμματισμός (Scheduling),
- συστήματα ελέγχου και παραγωγής.

Όλα τα παραπάνω βρίσκουν εφαρμογή στη Ρομποτική, στην Άμυνα, στην Αεροπορία και Αυτοκίνηση, στην Ιατρική Διαγνωστική, σε γεωλογικές έρευνες, στην Οικονομία και σε τραπεζικές εφαρμογές, στη Βιομηχανία και σε πολλούς ακόμα χώρους παραγωγής.

4.6 Γενετικοί Αλγόριθμοι

4.6.1 Φυσική εξέλιξη: Η αρχική έμπνευση

Οι **Γενετικοί Αλγόριθμοι-ΓΑ** (Genetic Algorithms-GA) αποτελούν μία από τις μεθόδους που χρησιμοποιεί η ΤΝ, για να δημιουργήσει συστήματα Μηχανικής Μάθησης που βασίζονται στην **Εξελικτική Μάθηση** (Evolutionary Learning). Πιο συγκεκριμένα, οι Γενετικοί Αλγόριθμοι είναι γνωστικά συστήματα που μιμούνται την αναπαραγωγική διαδικασία πληθυσμών απλών βιολογικών οργανισμών κατά τη φάση της φυσικής εξέλιξης. Η διαδικασία λαμβάνει χώρα στα **χρωμοσώματά** τους (chromosomes), τα οποία με τη σειρά τους καταγράφουν τη δομή κάθε οργανισμού μέσω των **γονιδίων** τους (genes).



Εικόνα 4.2 Δομή φυσικού χρωμοσώματος - <http://aliengrip.pblogs.gr/>

Μια ζωντανή ύπαρξη δημιουργείται μερικώς μέσω μιας διαδικασίας αποκωδικοποίησης των χρωμοσωμάτων. Οι ιδιαιτερότητες της κωδικοποίησης και αποκωδικοποίησης των χρωμοσωμάτων δεν έχουν κατανοηθεί πλήρως, αλλά μερικά από τα γενικά χαρακτηριστικά της θεωρίας είναι ευρέως αποδεκτά:

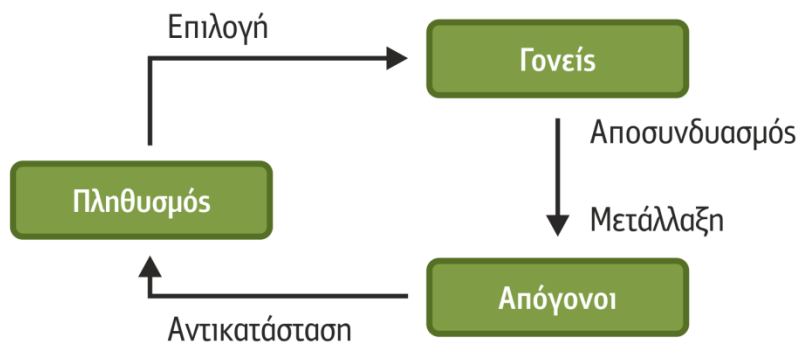
- Η φυσική επιλογή είναι ο σύνδεσμος μεταξύ των χρωμοσωμάτων και της απόδοσης των κωδικοποιημένων δομών. Διαδικασίες της φυσικής επιλογής επιτυγχάνουν την επικράτηση των χρωμοσωμάτων που περιέχουν τις πιο ισχυρές δομές και την αναπαραγωγή των πιο συχνά εμφανιζόμενων ισχυρών δομών.
- Κατά την αναπαραγωγή πραγματοποιείται η βιολογική εξέλιξη. Μεταλλάξεις μπορούν να κάνουν τα χρωμοσώματα των βιολογικών απογόνων να είναι διαφορετικά από αυτά των βιολογικών γονέων. Επίσης, και η διαδικασία του συνδυασμού χρωμοσωμάτων μπορεί να δημιουργήσει διαφορετικά χρωμοσώματα στους απογόνους, συνδυάζοντας “υλικά” από τα χρωμοσώματα των γονέων.
- Η βιολογική εξέλιξη δεν έχει μνήμη. Ό,τι γνωρίζει για την παραγωγή υπάρξεων που θα ταιριάζουν μέσα στο περιβάλλον τους εμπεριέχεται μέσα στον γενετικό χώρο, δηλαδή στο σύνολο των χρωμοσωμάτων από τις γονικές υπάρξεις, καθώς και στη δομή που τα χαρακτηρίζει.

Η συνδυασμένη δράση της φυσικής επιλογής και ο ανασυνδυασμός του γενετικού υλικού (των γονιδίων) κατά τη διάρκεια της αναπαραγωγής αποτελούν την κινητήριο δύναμη της εξέλιξης.

Η μίμηση των φυσικών μεθόδων οδήγησε στην ανάπτυξη των επονομαζόμενων **Εξελικτικών Αλγορίθμων** (Evolutionary Algorithms) μια κατηγορία άμεσων, πιθανολογικών αλγορίθμων αναζήτησης και βελτιστοποίησης.

Τυπικά, ένας εξελικτικός αλγόριθμος αρχικοποιεί τον πληθυσμό του σε τυχαίες τιμές, αν και μπορεί να χρησιμοποιηθεί προηγούμενη γνώση του πεδίου εφαρμογής (εάν υπάρχει), για να επιδράσει στην αρχικοποίηση του πληθυσμού. Ακολουθεί η *αξιολόγηση* του πληθυσμού με απόδοση αντίστοιχων τιμών ποιότητας σε κάθε άτομο του πληθυσμού στο συγκεκριμένο περιβάλλον. Η αξιολόγηση γίνεται μέσω της *συνάρτησης ποιότητας* (αντιπροσωπευτικής του συγκεκριμένου περιβάλλοντος), η οποία μπορεί να είναι πολύ απλή, όπως ο υπολογισμός μιας απλής συνάρτησης, ή εξαιρετικά πολύπλοκη, όπως η εκτέλεση μιας πολύπλοκης προσομοίωσης. Η *επιλογή* συνήθως υλοποιείται σε δύο βήματα: επιλογή γονέων και επιβίωση γονέων. Κατά την επιλογή των γονέων καθορίζεται ποια άτομα θα γίνουν γονείς και πόσους *απογόνους/παιδιά* (*offsprings/children*) θα αποκτήσουν.

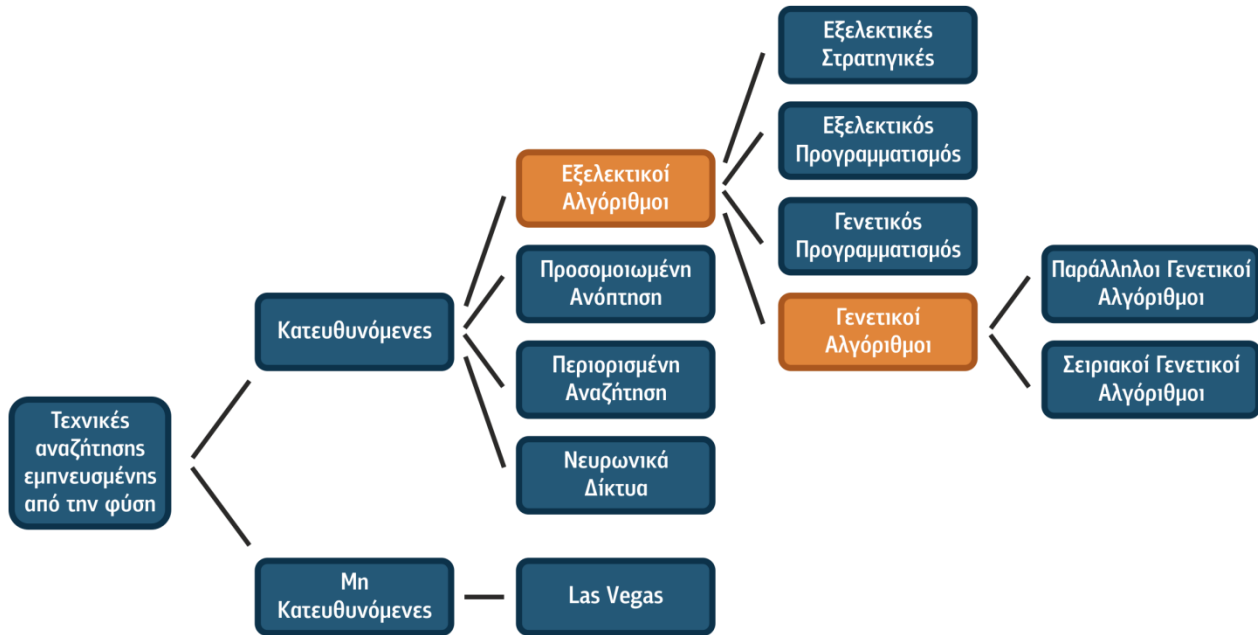
Οι απόγονοι δημιουργούνται μέσω **ανασυνδυασμού** των γονέων, δηλαδή με την ανταλλαγή πληροφορίας μεταξύ των γονέων και μέσω **μετάλλαξης** η οποία διαταράσσει περαιτέρω τους απογόνους. Ακολουθεί η χρήση της συνάρτησης ποιότητας για την αξιολόγηση των απογόνων και τελικά η επιλογή των ατόμων του πληθυσμού που θα επιβιώσουν στην επόμενη γενιά. Η διαδικασία αυτή ονομάζεται **εξελικτικός κύκλος** (evolutionary cycle). Το σχήμα 4.33 εμφανίζει τον συνήθη εξελικτικό κύκλο.



Σχήμα 4.33 Ο εξελικτικός κύκλος των Γενετικών Αλγορίθμων

4.6.2 Ιστορική Εξέλιξη

Ο όρος Γενετικός Αλγόριθμος (ΓΑ) χρησιμοποιήθηκε για πρώτη φορά από τον John Holland (1975) για έναν από τους βασικούς εκπροσώπους των εξελικτικών αλγορίθμων, ο οποίος θεωρείται και ως ο ευρύτερα γνωστός εξελικτικός αλγόριθμος των τελευταίων χρόνων. Όμως, πλέον, πολλοί ερευνητές χρησιμοποιούν κυρίως τον όρο **Εξελικτική Υπολογιστική** (Evolutionary Computation), προκειμένου να αναφερθούν στις εξελίξεις των τελευταίων ετών σε έναν τομέα που υπερβαίνει κατά πολύ τον αρχικό ΓΑ και ανήκει στις κατευθυνόμενες τεχνικές αναζήτησης εμπνευσμένες από τη φύση (βλέπε Σχήμα 4.34).



Σχήμα 4.34 Ο τομέας των μεθόδων αναζήτησης εμπνευσμένων από τη φύση

Αρκετοί επιστήμονες έχουν δραστηριοποιηθεί στην ανάπτυξη της Εξελκτικής Υπολογιστικής ήδη από τη δεκαετία του 1960. Το 1975, αφενός κυκλοφόρησε το βιβλίο του John Holland «*Adaptation in Natural and Artificial Systems*» και αφετέρου ο Kenneth DeJong, μεταπτυχιακός φοιτητής του Holland, ολοκλήρωσε τη διδακτορική του διατριβή με θέμα «*Analysis of the behavior of a class of genetic adaptive systems*». Θεωρείται ότι αμφότερα τα συγγράμματα έθεσαν τα θεμέλια των γενετικών αλγορίθμων. Ο DeJong ήταν, μάλιστα, ο πρώτος που παρέσχε μια ολοκληρωμένη παρουσίαση των δυνατοτήτων των ΓΑ στη βελτιστοποίηση και ακολούθησε μια σειρά από περαιτέρω μελέτες.

Η πρώτη διάσκεψη για αυτό το πρωτοεμφανισθέν ενδιαφέρον θέμα συγκλήθηκε το 1985. Ένας άλλος μεταπτυχιακός φοιτητής του Holland, ο David Goldberg, εκπόνησε τη βραβευμένη διδακτορική διατριβή του σχετικά με την εφαρμογή του ΓΑ στη βελτιστοποίηση αγωγού φυσικού αερίου. Κατόπιν, ο Goldberg εξέδωσε το 1989 το σημαντικό βιβλίο «*Genetic Algorithms in Search, Optimization and Machine Learning*», το οποίο λειτούργησε ως καταλύτης για την ανάπτυξη της θεωρίας και των εφαρμογών των ΓΑ, οι οποίες σήμερα εξακολουθούν να αυξάνονται με ταχείς ρυθμούς.

Σε αντίθεση με τους προηγούμενους εξελκτικούς αλγόριθμους, οι οποίοι επικεντρώνονταν στη μετάλλαξη των ατόμων του πληθυσμού και θα μπορούσαν να θεωρηθούν ως απλή εξέλιξη των μεθόδων αναρρίχησης λόφων, ο ΓΑ του Holland είχε ένα επιπλέον συστατικό, την ιδέα του ανασυνδυασμού των γονιδίων των γονέων.

4.6.3 Γενικά περί Γενετικών Αλγορίθμων

Η χρήση των Γενετικών Αλγορίθμων (ΓΑ) αποβλέπει στη δημιουργία απλών τρόπων αναζήτησης λύσεων σε προβλήματα, κυρίως βελτιστοποίησης και χρονοδρομολόγησης, τα οποία κατεξοχήν εξαρτώνται από μεγάλο πλήθος παραμέτρων, καθιστάμενα ελάχιστα κατανοητά, με πολλές αλληλεξαρτήσεις ή πολύ περίπλοκα ώστε δεν εφαρμόζονται η βελτιστοποίηση ή χρονοδρομολόγηση με άλλες πιο άμεσες μεθόδους.

Για να μπορέσει να τρέξει ένας ΓΑ, απαιτείται να έχουν ικανοποιηθεί 2 προϋποθέσεις επιτυχώς:

- μια κωδικοποίηση του χώρου επίλυσης και
- μια συνάρτηση καταλληλότητας που να αξιολογεί το χώρο επίλυσης.

Μόλις επιτευχθούν τα δυο παραπάνω, τότε μπορεί να εφαρμοστεί ο ΓΑ, για να δημιουργήσει και να αξιολογήσει τις υποψήφιες λύσεις μέσω μιας προσομοίωσης της εξελικτικής διαδικασίας. Οι συμβατικές μέθοδοι βελτιστοποίησης και οι μέθοδοι αναρρίχησης λόφου εντοπίζουν το πλησιέστερο τοπικό βέλτιστο μιας συνάρτησης, αλλά συνήθως αποτυγχάνουν στη εύρεση καθολικής λύσης. Από την άλλη, οι στατιστικές τεχνικές απαιτούν μεγάλο υπολογιστικό χρόνο, ενώ οι υπολογιστικές μέθοδοι απαιτούν την ύπαρξη παραγώγων, η χρήση των οποίων δεν συνιστάται σε μεγάλα πολυεπίπεδα συστήματα με αυξανόμενη πολυπλοκότητα και ασυνέχεια.

Αντίθετα, η χρήση των ΓΑ είναι αποτελεσματική τεχνική τυχαίας αναζήτησης για την εύρεση της βέλτιστης λύσης σε συγκεκριμένο χρόνο. Η τεχνική αυτή βασίζεται στην ανακάλυψη πρώτα πολλών τμηματικών λύσεων για τη δημιουργία του αρχικού πληθυσμού και μετά στο συνδυασμό των καλύτερων λύσεων μέχρι την επίτευξη της βέλτιστης εξ αυτών. Η όλη δυσκολία εντοπίζεται στην ορθή εκτίμηση της απόδοσης των τμηματικών λύσεων, διότι λανθασμένη εκτίμηση μπορεί να οδηγήσει σε λύσεις που θα απέχουν πολύ από τη ζητούμενη βέλτιστη.

Η λειτουργία των ΓΑ βασίζεται στη δυνατότητά τους να αναπαράγουν τον πληθυσμό σε γενεές. Κάθε γενεά προκύπτει από την προηγούμενη, αφού επιλεγούν οι γονείς και εφαρμοστούν οι γενετικοί τελεστές της διασταύρωσης και της μετάλλαξης.

Τα κυριότερα χαρακτηριστικά των ΓΑ είναι τα ακόλουθα:

- Είναι ένας νέος τρόπος προγραμματισμού, κατάλληλος για επίλυση προβλημάτων σε χώρους αναζήτησης ιδιαίτερα πολύπλοκους, προσαρμόσιμος σε ποικίλα και ευμετάβλητα περιβάλλοντα.
- Περιλαμβάνουν ένα σύνολο μεθόδων εμπνευσμένων από τη θεωρία της εξέλιξης. Αρχίζουν από έναν συνήθως επιλεγμένο πληθυσμό και επιδιώκουν την παραγωγή ατόμων με υψηλή απόδοση.
- Η προοδευτική βελτίωση του πληθυσμού επιτυγχάνεται με τις ενέργειες των γενετικών τελεστών που επιδρούν στην αναπαραγωγή, σύμφωνα με την ικανότητα των ατόμων, με εφαρμογή των τεχνικών της διασταύρωσης και της μετάλλαξης.
- Οι είσοδοι και οι έξοδοι του προβλήματος αναπαρίστανται ως συμβολοσειρές (strings) σταθερού μήκους, ενός συγκεκριμένου αλφαβήτου (συνήθως δυαδικού), με ρόλο αντίστοιχο του χρωμοσώματος στη Γενετική.
- Κάθε σημείο του χώρου επίλυσης του προβλήματος αντιπροσωπεύεται από μια συμβολοσειρά ή δομή, όπως αναφέρθηκε παραπάνω.
- Κάθε θέση στη συμβολοσειρά καταλαμβάνεται από ένα ή περισσότερα σύμβολα του αλφαβήτου, αντιπροσωπεύοντας μια μεταβλητή του προβλήματος και έχοντας ρόλο αντίστοιχο με αυτόν του γενετικού υλικού στους βιολογικούς μηχανισμούς, δηλαδή γονιδίου μέσα σε χρωμόσωμα.
- Απαιτείται η δυνατότητα πληροφόρησης για την ικανότητα των διάφορων συμβολοσειρών του πληθυσμού, σύμφωνα με τη **συνάρτηση καταλληλότητας** (fitness function), που καλείται επίσης και **συνάρτηση απόδοσης**, **συνάρτηση προσαρμογής** ή **αντικειμενική συνάρτηση**.
- Ο χρόνος στην εξέλιξη της διαδικασίας υπολογίζεται σε διακριτά διαστήματα τα οποία καλούνται *γενεές*: παράδειγμα $N(t)$: ο πληθυσμός κατά τη γενεά t .

4.6.4 Η Ανατομία ενός Γενετικού Αλγόριθμου

Τρία είναι τα βασικά λειτουργικά τμήματα ενός ΓΑ:

- **Τμήμα πληθυσμού**, με τις δομές και τεχνικές για τη δημιουργία και τη διαχείρισή του,
- **Τμήμα εκτίμησης της απόδοσης**, με τη συνάρτηση καταλληλότητας,
- **Τμήμα αναπαραγωγής**, με τεχνικές για τη δημιουργία νέων δομών με την εφαρμογή των γενετικών τελεστών.

Στα παραπάνω, οι βασικοί μηχανισμοί που συνδέουν τον ΓΑ με το πρόβλημα που επιλύει είναι ο τρόπος κωδικοποίησης της δομής που αναπαριστά μια λύση του προβλήματος, καθώς και η συνάρτηση καταλληλότητας που επιστρέφει τη μέτρηση της απόδοσης της δομής.

Επιπλέον, ο ΓΑ απαιτεί να οριστεί ένα **όριο καταλληλότητας**, το οποίο θα αντιστοιχεί στην επιθυμητή τιμή απόδοσης που πρέπει να επιτευχθεί από μια δομή, ώστε να τερματιστεί ο ΓΑ και 3 επιπλέον παράμετροι:

- το πλήθος N των δομών του αρχικού πληθυσμού,
- το ποσοστό r του πληθυσμού που θα αναπαραχθεί σε μια γενεά,
- το ποσοστό m των απογόνων που θα υποστούν μετάλλαξη.

Οι μηχανισμοί λειτουργίας του αλγορίθμου ποικίλλουν ως προς την εφαρμογή και τα αποτελέσματα.

Στο βιβλίο του «*Genetic Algorithms in Search, Optimization and Machine Learning*» (1989), ο Goldberg περιγράφει τι κάνει τους γενετικούς αλγόριθμους να λειτουργούν και εισάγει τον **απλό γενετικό αλγόριθμο** ή **κανονικό γενετικό αλγόριθμο** (canonical genetic algorithm): έναν εξελικτικό αλγόριθμο που εκτελεί ενέργειες σε μια συμβολοσειρά δυαδικών ψηφίων με τους ακόλουθους τρεις γενετικούς τελεστές:

- αναπαραγωγή
- διασταύρωση
- μετάλλαξη

Μια συνάρτηση καταλληλότητας αντιστοιχίζει τη δυαδική αναπαράσταση σε έναν πραγματικό αριθμό. Ο στόχος του κανονικού γενετικού αλγορίθμου είναι η βελτιστοποίηση της απόδοσης με την εξερεύνηση του δυαδικού χώρου αναζήτησης, ώστε ο πραγματικός αριθμός να ελαχιστοποιηθεί ή να μεγιστοποιηθεί.

Τα βήματα του Κανονικού Γενετικού Αλγόριθμου:

Δημιουργήστε έναν τυχαίο πληθυσμό με N δυαδικές συμβολοσειρές.

Αξιολογήστε την απόδοση της κάθε μεμονωμένης συμβολοσειράς μέσω της συνάρτησης καταλληλότητας $g(x)$.

Εφαρμόστε την αναπαραγωγή, δηλαδή επιλέξτε ζεύγη γονέων από τον πληθυσμό βάσει της απόδοσής τους.

Για κάθε ζεύγος γονέων επαναλάβετε:

Με πιθανότητα r εφαρμόστε τον τελεστή διασταύρωσης στους γονείς, για να παραχθεί μια νέα συμβολοσειρά-απόγονος.

Με πιθανότητα m εφαρμόστε τη μετάλλαξη σε κάποιο δυαδικό ψηφίο της συμβολοσειράς κάθε απόγονου ξεχωριστά.

Τοποθετήστε τη συμβολοσειρά-απόγονο στον νέο πληθυσμό.

Αντικαταστήστε τον παλιό με τον νέο πληθυσμό.

Αν δεν έχει βρεθεί αποδεκτή λύση ούτε έχει εξαντληθεί ο προκαθορισμένος μέγιστος αριθμός γενεών, μεταβαίνετε ξανά στο ανωτέρω βήμα 2.

4.6.5 Κωδικοποίηση

Στο περιβάλλον ενός ΓΑ, οι είσοδοι του προβλήματος είναι στην ουσία τα N χρωμοσώματα του αρχικού πληθυσμού που αντιπροσωπεύουν μερικές λύσεις του προβλήματος. Κάθε χρωμόσωμα αναπαρίσταται με δομή συμβολοσειράς (string) σταθερού μήκους n , ενός συγκεκριμένου αλφαβήτου. Τα χρωμοσώματα πρέπει να έχουν μία αναπαράσταση τέτοια, ώστε να περιέχουν μέσα τους τα χαρακτηριστικά της λύσης που εκπροσωπούν. Τα χαρακτηριστικά ενός χρωμοσώματος καλούνται **γονότυπος** (genotype) του χρωμοσώματος. Πάνω στον γονότυπο εφαρμόζονται οι γενετικοί τελεστές και παράγονται οι απόγονοι.

Ο Holland κωδικοποίησε το γονότυπο ενός χρωμοσώματος ως μία σειρά δυαδικών ψηφίων $\{0,1\}$. Η επιλογή του δυαδικού αλφαβήτου για την κωδικοποίηση διαθέτει έναν αριθμό θετικών χαρακτηριστικών που

επιτρέπουν τη δημιουργία απλών, αποδοτικών και κομψών ΓΑ. Όμως, υπάρχουν και άλλοι τρόποι για την κωδικοποίηση των χρωμοσωμάτων με δικά τους εγγενή πλεονεκτήματα. Για παράδειγμα, η κωδικοποίηση μέσα στο γονότυπο μπορεί να γίνει χρησιμοποιώντας ακέραιους ή πραγματικούς αριθμούς και άλλων αριθμητικών συστημάτων, όπως είναι το δεκαδικό, το οκταδικό, το δεκαεξαδικό κτλ.

Η συνάρτηση καταλληλότητας ενός χρωμοσώματος απαιτεί την αποκωδικοποίηση του γονότυπου στο δεκαδικό σύστημα. Η αποκωδικοποίηση αυτή οδηγεί στη δημιουργία του αντίστοιχου **φαινότυπου** (phenotype). Όταν η κωδικοποίηση γίνεται με το δυαδικό σύστημα αναπαράστασης, τότε για την αποκωδικοποίηση στο επίπεδο του φαινότυπου εφαρμόζεται μια απλή αντιστοίχιση της δυαδικής τιμής προς την ακέραια δεκαδική. Αντίστοιχα, αν η κωδικοποίηση γίνει με το δεκαδικό σύστημα, τότε υπάρχει συγχώνευση των επιπέδων του γονότυπου και του φαινότυπου.

Παράδειγμα δυαδικού γονότυπου και αντίστοιχου φαινότυπου:

Γονότυπος: 01101

Φαινότυπος: 13

Κάθε σημείο (λύση) του πεδίου του προβλήματος αποτελεί μια δομή (χρωμόσωμα ή άτομο) του πληθυσμού που έχει αναπαρασταθεί μέσω του επιλεγμένου αλφαβήτου. Σε οποιαδήποτε χρονική στιγμή, το όλο σύστημα περιέχει έναν πληθυσμό από N άτομα που αναπαριστούν το σύνολο των μέχρι τότε εντοπισμένων λύσεων στο πρόβλημα.

4.6.6 Η αρχικοποίηση ενός πληθυσμού

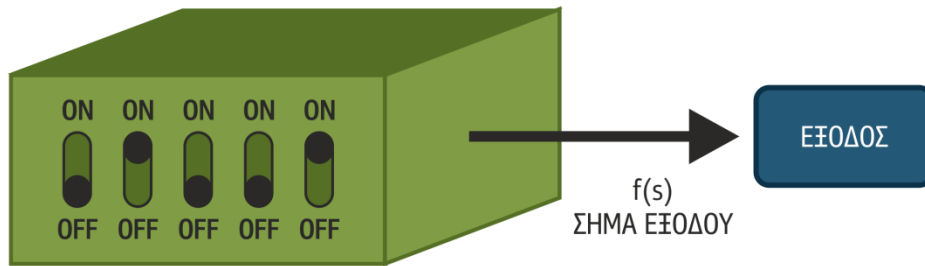
Το πρώτο βήμα ενός ΓΑ είναι να αρχικοποιήσει έναν πληθυσμό από πιθανές λύσεις του προβλήματος σε μορφή συμβολοσειρών αποδεκτού γονότυπου κωδικοποιημένου βάσει ενός επιλεγμένου αλφαβήτου.

Η αρχικοποίηση ενός τυχαίου πληθυσμού είναι η πλέον συνηθισμένη τακτική. Ως **τυχαίο πληθυσμό** εννοούμε ένα πλήθος δομών των οποίων ο γονότυπος καθορίζεται από μια τυχαία διαδικασία. Ο αρχικός πληθυσμός χαρακτηρίζεται από το πλήθος N των συμβολοσειρών μέσα σε αυτόν και από το μήκος των συμβολοσειρών του που είναι πάντα σταθερό.

Το πλήθος N των συμβολοσειρών μέσα σε έναν πληθυσμό χαρακτηρίζεται ως **μέγεθος του πληθυσμού** και μπορεί να παραμένει σταθερό ή να μεταβάλλεται κατά την εξέλιξη του πληθυσμού. Το μέγεθος του αρχικού πληθυσμού ορίζεται από μια παράμετρο που δίνεται κατά την αρχικοποίηση του ΓΑ και η οποία πρέπει να επιλέγεται με προσοχή, γιατί ανάλογα με το κάθε πρόβλημα, το βέλτιστο μέγεθος διαφέρει. Όταν το μέγεθος του πληθυσμού είναι πολύ μικρό, τότε δεν γίνεται επαρκής εξερεύνηση του συνολικού χώρου αναζήτησης, δεδομένου ότι η σύγκλιση είναι ταχύτερη. Εάν το μέγεθος του πληθυσμού είναι πολύ μεγάλο, τότε ο χρόνος θα αναλωθεί με την επεξεργασία περισσότερων δεδομένων από όσα απαιτούνται και οι χρόνοι σύγκλισης θα είναι σημαντικά μεγαλύτεροι.

Παράδειγμα αρχικοποίησης πληθυσμού

Έστω το πρόβλημα του Μαύρου Κουτιού με τους 5 διακόπτες εισόδου, των οποίων την έξοδο επιθυμούμε να βελτιστοποιήσουμε. Ο κάθε διακόπτης μπορεί να βρίσκεται στην κατάσταση off ή on που αντιστοιχεί στην κατάσταση 0 ή 1 (βλέπε Σχήμα 4.35).



Σχήμα 4.35 Το πρόβλημα βελτιστοποίησης του Μαύρου Κουτιού με τους 5 διακόπτες

Μια λύση του προβλήματος αναπαρίσταται ως ένα χρωμόσωμα με γονότυπο 5 γονιδίων, τα οποία παρουσιάζουν την κατάσταση στην οποία βρίσκεται καθένας από τους 5 διακόπτες τη δεδομένη στιγμή. Η κωδικοποίηση του γονότυπου γίνεται με το δυαδικό αλφάβητο και η συμβολοσειρά που προκύπτει είναι ένας 5ψήφιος δυαδικός αριθμός. Παρακάτω, δίνεται ένας τυχαία δημιουργημένος αρχικός πληθυσμός μεγέθους 4, όπου κάθε δομή αντιστοιχεί σε έναν συνδυασμό καταστάσεων των διακοπών (όπου 0=κλειστός και 1=ανοιχτός):

```
01101
11000
01000
10011
```

Το παραπάνω παράδειγμα προέρχεται από το βιβλίο του Goldberg «*Genetic Algorithms in Search, Optimization & Machine Learning*» και θα χρησιμοποιηθεί και σε επόμενες παραγράφους του κεφαλαίου.

4.6.7 Θεωρία σχημάτων

Ένα λεπτό σημείο που πρέπει να προσεχθεί στην αναπαράσταση ενός χρωμοσώματος μέσω μιας δομής είναι τα δομικά της κομμάτια που ονομάζονται και δυνατά σχήματα της δομής του χρωμοσώματος.

Ένα **σχήμα** (schema) ή αλλιώς **πρότυπο ομοιότητας** (prototype resemblance) μπορεί ουσιαστικά να οριστεί ως ένα πρότυπο χρωμοσωμάτων.

Για παράδειγμα, ας σκεφτούμε έναν πληθυσμό χρωμοσωμάτων στον οποίο το καθένα από αυτά, δηλαδή η καθεμία υποψήφια λύση, έχει την κωδικοποίηση που παρουσιάστηκε στο παράδειγμα της προηγούμενης παραγράφου.

Έστω ότι 01101 και 01010 είναι δύο τέτοια χρωμοσώματα. Μια κοινή περιγραφή για αυτά τα χρωμοσώματα είναι «τα χρωμοσώματα των οποίων τα δύο πρώτα bits είναι 01». Έτσι, υποστηρίζουμε ότι το 01*** είναι ένα πρότυπο, δηλαδή ένα σχήμα καθορισμού αυτών των χρωμοσωμάτων. Οι αστερίσκοι αντιπροσωπεύουν «οποιαδήποτε τιμή», κάτι το οποίο σημαίνει ότι μπορεί να είναι είτε 0 είτε 1. Τα 0 και τα 1 ονομάζονται σταθερά bits. Ως εκ τούτου, ένα σχήμα στο δυαδικό αλφάβητο είναι ένα πρότυπο το οποίο χρησιμοποιεί τα σύμβολα {0, 1, *}, για να καθορίσει ένα σύνολο χρωμοσωμάτων. Εύκολα μπορούμε να συμπεράνουμε ότι ένα σχήμα αντιπροσωπεύει 2^n χρωμοσώματα στον πληθυσμό, όπου n είναι ο αριθμός των αστερίσκων στο σχήμα. Στο παράδειγμά μας, το 01*** αντιπροσωπεύει ένα σύνολο από 8 χρωμοσώματα, δεδομένου ότι υπάρχουν τρεις αστερίσκοι στο σχήμα, το [01100, 01000, 01110, 01111, 01001, 01010, 01011, 011101]. Προφανώς, το ***** αντιπροσωπεύει ολόκληρο τον πληθυσμό.

Σύμφωνα με τη Θεωρία Σχημάτων, ως **μήκος ορισμού** (defining length) ενός σχήματος ορίζεται η διαφορά των θέσεων του πρώτου και του τελευταίου καθορισμένου bit μέσα στο σχήμα. Για παράδειγμα, για το σχήμα 01***01***, το πρώτο καθορισμένο bit (το πρώτο 0) είναι στη θέση 1 και το τελευταίο καθορισμένο bit (το δεύτερο 1) είναι στη θέση 7. Έτσι, το σχήμα αυτό έχει μήκος $7 - 1 = 6$. Η **τάξη** (order) ενός σχήματος είναι ο αριθμός των καθορισμένων του bits. Το 01***10*** έχει 4 σταθερά καθορισμένα ψηφία, δηλαδή έχει τάξη 4.

Η Θεωρία Σχημάτων εξηγεί ποιο συγκεκριμένο σχήμα είναι πιθανόν να διαδοθεί από γενεά σε γενεά:

- Τα σχήματα με μήκος κοντά στο μήκος του χρωμοσώματος έχουν λιγότερες πιθανότητες να επιβιώσουν λόγω της διασταύρωσης.
- Τα σχήματα υψηλής τάξης, δηλαδή με πολλά καθορισμένα bits, είναι επίσης πιο πιθανό να εξαφανιστούν ως αποτέλεσμα της μετάλλαξης.
- Συνάγεται το συμπέρασμα ότι τα μικρού μήκους, χαμηλής τάξης σχήματα που έχουν καταλληλότητα πάνω από τον μέσο όρο είναι πιο πιθανόν να υπάρχουν στις διάδοχες γενεές. Τέτοια σχήματα ονομάζονται δομικά στοιχεία και μπορούν να οριστούν ως επιμέρους λύσεις. Ένα χρωμόσωμα το οποίο έχει πολλά δομικά στοιχεία έχει μεγάλες πιθανότητες να προσεγγίζει μια σχεδόν βέλτιστη λύση και στη διαδικασία επιλογής πρέπει να του δίνονται περισσότερες ευκαιρίες διάδοσης στην επόμενη γενεά.

4.6.8 Αναπαραγωγή

Οι τυχαίοι αρχικοί πληθυσμοί είναι σχεδόν πάντα εξαιρετικά αποτυχημένοι στο να προσφέρουν ικανοποιητικές λύσεις. Οι λύσεις που περιέχονται σε μια γενεά και αποτελούν τα χρωμοσώματα του πληθυσμού της πρέπει να αξιολογηθούν, πριν τους επιτραπεί να συμμετέχουν στις διαδικασίες αναπαραγωγής της επόμενης γενεάς με στόχο τη βελτίωση του πληθυσμού. **Αναπαραγωγή** (Reproduction) σε ένα ΓΑ καλείται η αξιολόγηση των δομών του πληθυσμού μιας γενεάς με τη βοήθεια της συνάρτησης καταλληλότητας, για να τους δοθεί στη συνέχεια η ευκαιρία, μικρότερη ή μεγαλύτερη ανάλογα με την απόδοσή τους, να συμμετέχουν μέσω διαδικασιών αναπαραγωγής στη δημιουργία του πληθυσμού της επόμενης γενεάς.

Ανάλογα με τον τρόπο με τον οποίο δομούμε τη συνάρτηση καταλληλότητας, μπορούμε να στοχεύουμε στη δημιουργία είτε του λιγότερο δαπανηρού πληθυσμού είτε του περισσότερο κατάλληλου, θέτοντας κατ' αντιστοιχία ένα ζήτημα ελαχιστοποίησης του κόστους ή μεγιστοποίησης της φυσικής καταλληλότητας. Στα προβλήματα βελτιστοποίησης, το κόστος δεν είναι μέτρο της αξίας (π.χ. χρηματικής), αλλά ένα μέτρο αποδοτικότητας. Φυσικά, η καταλληλότητα μπορεί να θεωρηθεί ως αντιστρόφως ανάλογη με το κόστος• ως εκ τούτου, μπορεί να γίνει εύκολα μετατροπή από το ένα μέγεθος στο άλλο. Ο σχεδιασμός μιας συνάρτησης καταλληλότητας που μπορεί να αποτιμήσει επιτυχώς την αξία ενός πληθυσμού βάσει της διαθέσιμης γνώσης για το πρόβλημα είναι βασική προϋπόθεση της επιτυχίας ενός ΓΑ και αποτελεί ένα από τα δυσκολότερα προβλήματα που έχει να αντιμετωπίσει ο δημιουργός του.

Στις τεχνικές βελτιστοποίησης, το εύρος των πιθανών λύσεων αναφέρεται συχνά ως ο χώρος των λύσεων, ενώ η καταλληλότητα του κάθε σημείου στο χώρο λύσεων αναφέρεται ως το **υψόμετρο** (altitude) στο τοπίο του προβλήματος. Η αναζήτηση για το ολικό ελάχιστο (ή μέγιστο) του κόστους είναι το ίδιο με την αναζήτηση του χαμηλότερου σημείου στη χαμηλότερη κοιλάδα του τοπίου του κόστους. Ομοίως, η αναζήτηση για το ολικό μέγιστο της καταλληλότητας είναι το ίδιο με την αναζήτηση για το υψηλότερο σημείο του ψηλότερου όρους στο τοπίο της καταλληλότητας.

Γονική επιλογή

Σκοπός της γονικής επιλογής σε ένα ΓΑ είναι να δοθούν περισσότερες πιθανότητες αναπαραγωγής σε εκείνα τα μέλη του πληθυσμού που είναι ισχυρότερα βάσει της εκτίμησης της συνάρτησης καταλληλότητας. Τα μέλη που επιλέγονται ως κατάλληλα για αναπαραγωγή καλούνται γονείς.

Ο κλασικός μηχανισμός επιλογής γονέων είναι η **ρουλέτα γονικής επιλογής** (roulette wheel selection), η οποία επιλέγει μέλη ενός πληθυσμού με πιθανότητα ανάλογη προς την καταλληλότητά τους με βάση το μέτρο της απόδοσής τους.

Τα βήματα του αλγόριθμου της Ρουλέτας Γονικής Επιλογής

- Άθροισε την απόδοση όλων των μελών του πληθυσμού. Κάλυψε το άθροισμα συνολική απόδοση.
- Δημιούργησε έναν τυχαίο αριθμό n μεταξύ 0 και της συνολικής απόδοσης.

- Επίστρεψε το πρώτο μέλος του πληθυσμού του οποίου η απόδοση προστιθέμενη στην απόδοση των προηγούμενων μελών είναι $\geq n$.
- Επανάλαβε τη διαδικασία τόσες φορές όσες ο πληθυσμός.

Ο αλγόριθμος αναφέρεται ως *ρουλέτα γονικής επιλογής*, γιατί μπορεί να παρομοιαστεί με την περιστροφή ενός τροχού ρουλέτας χωρισμένου σε τόσους κυκλικούς τομείς όσα τα μέλη του πληθυσμού, όπου κάθε τομέας αντιστοιχεί σε ένα διαφορετικό μέλος του πληθυσμού και είναι ανάλογος σε εμβαδόν με την απόδοση του μέλους. Η επιλογή ενός μέλους ως γονέα μπορεί να απεικονιστεί ως μια περιστροφή του τροχού με επιλεγμένο το γονέα στον τομέα του οποίου θα σταματήσει ο τροχός.

Παράδειγμα εφαρμογής ρουλέτας γονικής επιλογής

Στον πληθυσμό του προηγούμενου παραδείγματος της [παραγράφου 4.6.6](#) εφαρμόζουμε τη συνάρτηση καταλληλότητας g , όπου $g(x) = x^2$, η οποία δίνει για κάθε μέλος του πληθυσμού τις τιμές απόδοσης (πίνακας 4.5) και την αντίστοιχη πιθανότητα κάθε μέλους να επιλεγεί ως γονέας της νέας γενεάς.

A/A	ΜΕΛΟΣ	ΤΙΜΗ ΑΠΟΔΟΣΗΣ	ΤΟΜΕΑΣ ΠΟΥ ΑΝΤΙΣΤΟΙΧΕΙ	ΠΙΘΑΝΟΤΗΤΑ ΕΠΙΛΟΓΗΣ %
1	01101	169	1 ^{ος} (0-169)	14,4
2	11000	576	2 ^{ος} (170-745)	49,2
3	01000	64	3 ^{ος} (746-809)	5,5
4	10011	361	4 ^{ος} (810-1170)	30,9
ΣΥΝΟΛΟ:		1170		100,0

Πίνακας 4.5 Πληθυσμός προς αναπαραγωγή και οι αντίστοιχες τιμές απόδοσης των μελών του

Οι τιμές απόδοσης υπολογίστηκαν με την εφαρμογή της συνάρτησης g σε κάθε αποκωδικοποιημένη δυαδική συμβολοσειρά στο δεκαδικό σύστημα, δηλαδή στο φαινότυπο κάθε χρωμοσώματος του πληθυσμού.

Παράδειγμα υπολογισμού για το μέλος 01101:

Γονότυπος: 01101

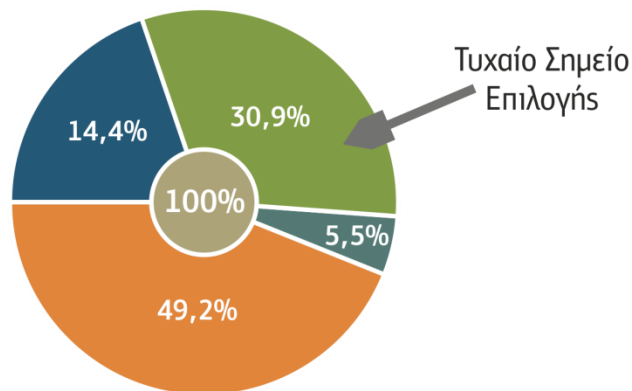
Φαινότυπος: 13

Υπολογισμός τιμής απόδοσης: $g(13) = 13^2 = 169$

Υπολογισμός πιθανότητας επιλογής (P): $P = \frac{169}{1170} * 100 = 14.4$

Ομοίως υπολογίζονται οι τιμές απόδοσης και οι πιθανότητες επιλογής για τα υπόλοιπα 3 μέλη του πληθυσμού.

Σύμφωνα με τα ποσοστά που εμφανίζονται στον παραπάνω πίνακα 4.5, προκύπτει η ρουλέτα του σχήματος 4.36, η οποία θα χρησιμοποιηθεί για την επιλογή των γονέων και την αναπαραγωγή της επόμενης γενεάς:



■ Τομέας 1 ■ Τομέας 2 ■ Τομέας 3 ■ Τομέας 4

Σχήμα 4.36 Ρουλέτα γονικής επιλογής

Για να ενεργοποιηθεί η διαδικασία της αναπαραγωγής, απλώς τρέχουμε επαναληπτικά τα βήματα του αλγόριθμου της ρουλέτας 4 φορές (όσες και το μέγεθος του πληθυσμού).

Βήματα Αλγόριθμου Ρουλέτας Γονικής Επιλογής:

- Άθροισε την απόδοση όλων των μελών του πληθυσμού.
- Κάλυψε το άθροισμα **συνολική απόδοση**.
- Δημιούργησε έναν **τυχαίο αριθμό n** μεταξύ 0 και της συνολικής απόδοσης.
- Επίστρεψε το πρώτο μέλος του πληθυσμού του οποίου η απόδοση, προστιθέμενη στην απόδοση των προηγούμενων μελών, είναι $\geq n$.
- Επανάλαβε τη διαδικασία τόσες φορές όσες ο πληθυσμός.

Στο συγκεκριμένο παράδειγμα, το πρώτο μέλος έχει τιμή απόδοσης ίση με 169, που αντιστοιχεί στο 14.4% της συνολικής απόδοσης. Ως αποτέλεσμα, στο συγκεκριμένο μέλος αποδίδεται το 14.4% του εμβαδού της ρουλέτας που αντιστοιχεί σε ένα διάστημα που αρχίζει από το 0 και τελειώνει στο 169. Κάθε περιστροφή της ρουλέτας θα επιλέξει το πρώτο μέλος του πληθυσμού, αν ο τροχός επιστρέψει μια τιμή από 0-169, με πιθανότητα να συμβεί 0.144. Το δεύτερο μέλος έχει τιμή απόδοσης ίση με 576, που αντιστοιχεί στο 49.2% της συνολικής απόδοσης. Ως αποτέλεσμα, στο συγκεκριμένο μέλος αποδίδεται το επόμενο διάστημα που αρχίζει από το 170 και τελειώνει στο 745 (170+ 576) που αντιστοιχεί στο 49.2% του εμβαδού της ρουλέτας. Μια περιστροφή της ρουλέτας θα επιλέξει το δεύτερο μέλος του πληθυσμού, αν ο τροχός επιστρέψει μια τιμή από 170-745, με πιθανότητα να συμβεί 0,492. Τα αντίστοιχα συμβαίνουν για τα υπόλοιπα δύο μέλη.

Εφαρμόζοντας τη ρουλέτα στον πληθυσμό του παραδείγματός μας έχουμε ως πιθανά αποτελέσματα αυτά που αποτυπώνονται στον Πίνακα 4.6.

Μέλη Πληθυσμού	01101 (Τομέας 1)	11000 (Τομέας 2)	01000 (Τομέας 3)	10011 (Τομέας 4)
Αποδόσεις ($g(x)=x^2$)	169	576	64	361
Προοδευτικό Άθροισμα Αποδόσεων	169	745	809	1170
Τυχαίος Αριθμός n	150	343	610	1000
Τομέας Ρουλέτας	1	2	2	4
Επιλεγμένος Γονέας	01101	11000	11000	10011

Πίνακας 4.6 Αποτελέσματα ρουλέτας γονικής επιλογής

Δείτε [κινούμενη εικόνα 4.5 - Ρουλέτα γονικής επιλογής](#)

Στον παρακάτω πίνακα 4.7 παρουσιάζονται όλα τα μεγέθη που υπολογίστηκαν για την παραγωγή των γονέων του παραδείγματός μας με χρήση της ρουλέτας γονικής επιλογής

A/A	Αρχικός πληθυσμός τυχαία δημιουργημένος	Δεκαδική τιμή x (φαινότυπος)	$g(x)=x^2$	Πιθανότητα επιλογής $g(x)/\sum_i g(i)$	Αναμ/νος αριθμός επιλογών $g(x)/\bar{g}^*$	Πραγματικός αριθμός επιλογών
1	01101	13	169	0,14	0,85	1
2	11000	24	576	0,49	1,97	2
3	01000	8	64	0,06	0,22	0
4	10011	19	361	0,31	1,23	1
	Άθροισμα		1170	1,00	4,00	4,0
	Μέσος όρος		293	0,25	1,00	1,0
	Μέγιστο		576	0,49	1,97	2,0

* όπου \bar{g} η μέση απόδοση του πληθυσμού

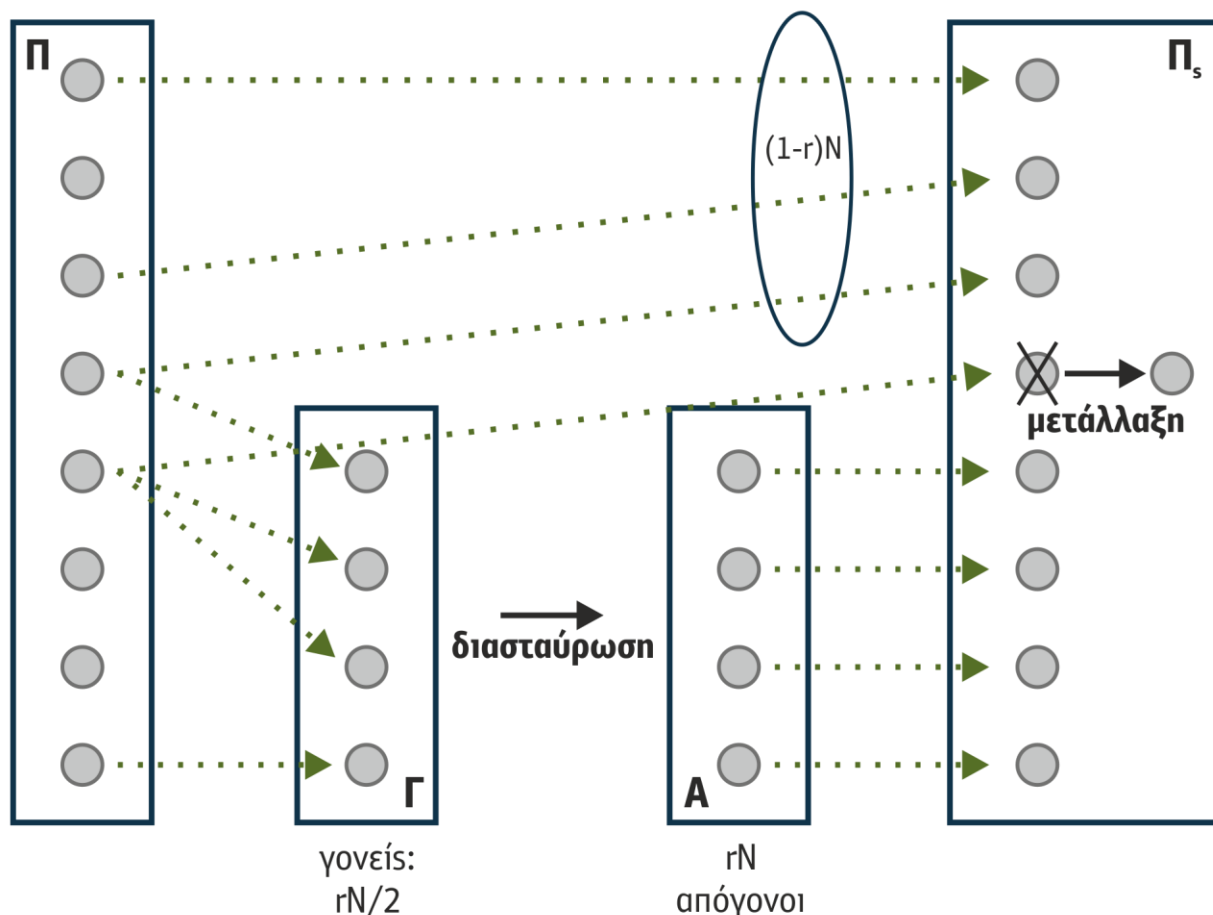
Πίνακας 4.7 Αποτύπωση όλων των στοιχείων για την επιλογή γονέων μέσω ρουλέτας γονικής επιλογής

Η μέθοδος της ρουλέτας ευνοεί την επιλογή μελών με τη μεγαλύτερη απόδοση, δίνοντάς τους μεγαλύτερη πιθανότητα επιλογής, δεδομένου ότι το εμβαδόν του κυκλικού τομέα τους είναι μεγαλύτερο και ως εκ τούτου ευνοείται η επιλογή του.

Μια ακόμη πολύ συχνά χρησιμοποιούμενη τεχνική είναι η **επιλογή τουρνουά** (tournament selection). Σε γενικές γραμμές, η διαδικασία επιλογής τουρνουά επιλέγει τυχαία μια ομάδα από μέλη του τρέχοντος πληθυσμού, συγκρίνει τις αποδόσεις τους και επιλέγει εκείνο το οποίο έχει την υψηλότερη απόδοση ως γονέας για την επόμενη γενεά. Η επιλογή τουρνουά θεωρείται ως μία από τις ταχύτερες μεθόδους επιλογής, είναι εύκολα υλοποιήσιμη και προσφέρει καλό έλεγχο της επιλεκτικής πίεσης ([βλέπε παρακάτω](#)).

Ανάλογα με τον ΓΑ, στη διαδικασία επιλογής γονέων μπορεί να συμμετέχει το σύνολο των μελών του πληθυσμού (N) ή μόνο ένα μέρος του (rN). Οι γονείς που θα επιστρέψει ο μηχανισμός επιλογής αποτελούν έναν προσωρινό πληθυσμό, στον οποίο θα επιδράσει ο γενετικός τελεστής της διασταύρωσης, για να παραχθεί ένας πληθυσμός rN απογόνων από rN/2 ζεύγη γονέων. Ο προσωρινός πληθυσμός μπορεί να συμπληρωθεί με (1-r)N μέλη της παλαιάς γενεάς, κατάλληλα επιλεγμένα με βάση το μηχανισμό επιλογής του ΓΑ σε περίπτωση κατά την οποία το πλήθος του είναι μικρότερο από το μέγεθος του πληθυσμού (r≠1). Ο

συμπληρωμένος προσωρινός πληθυσμός θα αποτελέσει τη νέα γενεά, αφού πρώτα επιδράσει σε αυτόν ο γενετικός τελεστής της μετάλλαξης (βλέπε Σχήμα 4.37).



Σχήμα 4.37 Αναπαραγωγή πληθυσμού

4.6.9 Γενετικοί Τελεστές

Απαραίτητοι για την εξέλιξη του πληθυσμού είναι οι **γενετικοί τελεστές** (genetic operators), που επιδρούν στον πληθυσμό επαναληπτικά, με απώτερο σκοπό να περιληφθούν σε αυτόν οι βέλτιστες λύσεις ή δομές, δηλαδή οι συμβολοσειρές με την καλύτερη τιμή απόδοσης.

Ένας ΓΑ αποδίδει ικανοποιητικά αποτελέσματα βάσει δύο θεμελιωδών γενετικών τελεστών:

- Διασταύρωση
- Μετάλλαξη

Οι τελεστές υποστηρίζουν ο ένας τον άλλον κατά σημαντικό τρόπο. Η μετάλλαξη εισάγει τη διαφοροποίηση στον πληθυσμό και η διασταύρωση ενεργεί για το σχεδιασμό των καλών σχημάτων που ήδη υπάρχουν. Αν η μετάλλαξη καταργηθεί, ο αλγόριθμος συγκλίνει γρήγορα, ενώ λείπουν σημαντικά σχήματα. Όταν τα σχήματα δεν είναι πλήρη, η διασταύρωση δεν μπορεί να κάνει πραγματική διαφοροποίηση στον πληθυσμό. Αν αφαιρεθεί η διασταύρωση, το σύστημα χρησιμοποιεί τη μετάλλαξη για την ανίχνευση μιας

συγκεκριμένης περιοχής χωρίς τη δυνατότητα σχεδιασμού καλύτερων χαρακτηριστικών των σημείων της περιοχής αυτής, που οδηγεί αυτονόητα το σύστημα σε μειωμένη απόδοση.

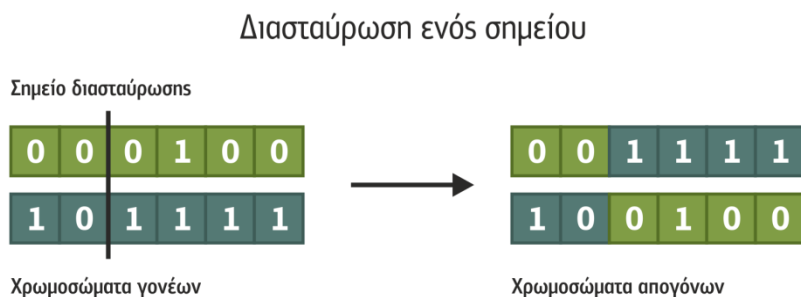
4.6.10 Διασταύρωση

Αφού επιλέγουν οι γονείς, η αναπαραγωγή αυτή καθεαυτή μπορεί να ξεκινήσει εφαρμόζοντας καταρχάς τον γενετικό τελεστή της **Διασταύρωσης** (Crossover) που στη βιολογία αναφέρεται και ως **Γενετική Μείωση**. Η αναπαραγωγή μέσω διασταύρωσης είναι συνδυασμένη πληροφορία από δυο γονείς για το σχηματισμό απογόνων. Πολλοί ερευνητές πιστεύουν πως η διασταύρωση είναι τόσο σημαντικό συστατικό ενός ΓΑ, ώστε, αν αφαιρεθεί από αυτόν, τότε παύει να είναι γενετικός αλγόριθμος. Ο μηχανισμός της αναπαραγωγής μαζί με τη διασταύρωση έχουν ως αποτέλεσμα τα καλύτερα σχήματα να επικρατούν στον πληθυσμό και ο συνδυασμός τους να παράγει νέα σχήματα υψηλής ποιότητας.

Μια νέα δομή παράγεται επιλέγοντας είτε από τον ένα γονέα είτε από τον άλλον ένα *αλληλόμορφο* (*allele*) για καθένα από τα γονίδια του χρωμοσώματος. Η διαδικασία συνδυασμού των γονιδίων μπορεί να πραγματοποιηθεί με διάφορους τρόπους που παρουσιάζονται στη συνέχεια.

Διασταύρωση Ενός Σημείου

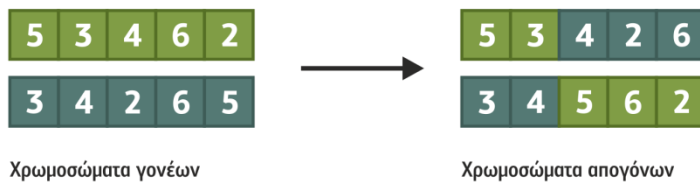
Η απλούστερη μέθοδος συνδυασμού ονομάζεται **διασταύρωση ενός σημείου** (single point crossover), την οποία εδώ παρουσιάζουμε χρησιμοποιώντας γονίδια που κωδικοποιούνται δυαδικά (βλέπε Σχήμα 4.38), αν και η διαδικασία βρίσκει εφαρμογή σε σχεδόν οποιαδήποτε γονιδιακή αναπαράσταση (βλέπε Σχήμα 4.39). Το σημείο διασταύρωσης επιλέγεται τυχαία κάπου στη σειρά των γονιδίων. Όλο το γενετικό υλικό μετά από το σημείο διασταύρωσης λαμβάνεται από τον ένα γονέα και όλο το υλικό μετά το σημείο διασταύρωσης λαμβάνεται από τον άλλο.



Σχήμα 4.38 Διασταύρωση ενός σημείου

Στην περίπτωση κωδικοποίησης με αλφάβητο φυσικών αριθμών, η διασταύρωση ενός σημείου θα μπορούσε να επιτευχθεί με μετάθεση των γονιδίων ενός γονέα με οδηγό τον άλλο γονέα, όπως στο παράδειγμα του σχήματος 4.39, όπου ο πληθυσμός έχει χρωμοσώματα μήκους 5 με γονίδια κωδικοποιημένα στο αλφάβητο {2 3 4 5 6}, με κάθε γονίδιο του να έχει διαφορετική τιμή από τα άλλα. Αν το σημείο διασταύρωσης είναι το 2^ο και ο 1^{ος} γονέας είναι ο (5,3,4,6,2) και ο 2^{ος} ο (3,4,2,6,4), τότε ο ένας απόγονος διαμορφώνεται από τα πρώτα δύο γονίδια του 1^{ου} γονέα, δηλαδή τα 5 & 3, και τα υπόλοιπα 3 που υπολείπονται, δηλαδή τα 4,2 & 6, προέρχονται από τον 2^ο γονέα και τοποθετούνται στη σειρά με την οποία εμφανίζονται στον 2^ο γονέα. Το αντίστοιχο συμβαίνει για τον άλλο απόγονο.

Διασταύρωση ενός σημείου για φυσικούς αριθμούς



Σχήμα 4.39 Διασταύρωση ενός σημείου για φυσικούς αριθμούς

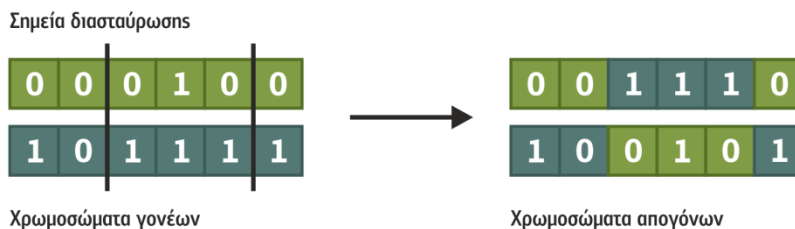
Διασταύρωση Πολλαπλών Σημείων

Η διασταύρωση ενός σημείου έχει το μειονέκτημα ότι στην αναπαράσταση που χρησιμοποιείται στον αλγόριθμο τα αλληλόμορφα που βρίσκονται κοντά μεταξύ τους διατηρούνται μαζί με πολύ μεγαλύτερη πιθανότητα από εκείνα που είναι περισσότερο απομακρυσμένα.

Ένας απλός τρόπος για τη μείωση αυτού του προβλήματος είναι να χρησιμοποιηθεί μια διαδικασία διασταύρωσης δύο σημείων. Στη **διασταύρωση δύο σημείων** (two-points crossover) δημιουργούνται δύο τυχαίες θέσεις. Στους απογόνους αντιγράφονται από τον ένα γονέα τα αλληλόμορφα πριν και μετά τα σημεία διασταύρωσης, και από τον άλλο γονέα αυτά που είναι ανάμεσα στα σημεία διασταύρωσης.

Η διαδικασία της διασταύρωσης μπορεί να πραγματοποιηθεί με περισσότερα από δύο σημεία διασταύρωσης. Ακόμη και όλα τα σημεία μπορούν να επιλεγούν για διασταύρωση, αν το επιθυμούμε. Σε αυτή την περίπτωση, κάθε γονίδιο έχει ίση πιθανότητα να προέλθει από τον ένα ή τον άλλο γονέα και μπορεί να υπάρχει σημείο διασταύρωσης μετά από οποιοδήποτε γονίδιο. Αυτή η ιδέα οδηγεί στη γενίκευση ότι θα μπορούσε κανείς, για παράδειγμα, να χρησιμοποιήσει έναν τυχαίο αριθμό σημείων διασταύρωσης σε κάθε επανάληψη.

Διασταύρωση δύο σημείων

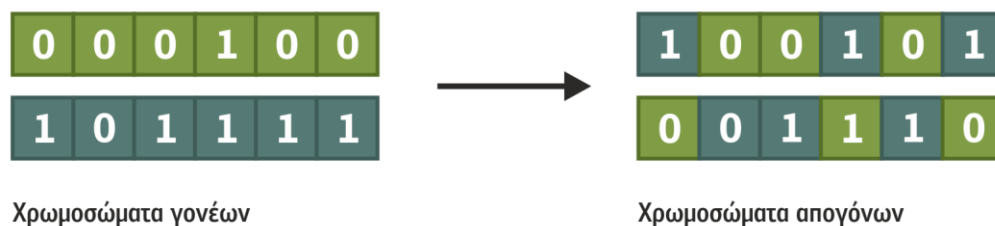


Σχήμα 4.40 Διασταύρωση δυο σημείων

Ομοιόμορφη Διασταύρωση

Μία μέθοδος διασταύρωσης, που χρησιμοποιείται συχνά, είναι η **ομοιόμορφη διασταύρωση** (uniform crossover), η οποία λειτουργεί ως εξής: Αρχικά επιλέγονται δύο γονείς με στόχο να δημιουργηθούν δύο απόγονοι. Για κάθε γονιδιακή θέση ενός απογόνου, αποφασίζεται τυχαία ποιος από τους γονείς θα συνεισφέρει το δικό του αλληλόμορφο που βρίσκεται στην αντίστοιχη θέση.

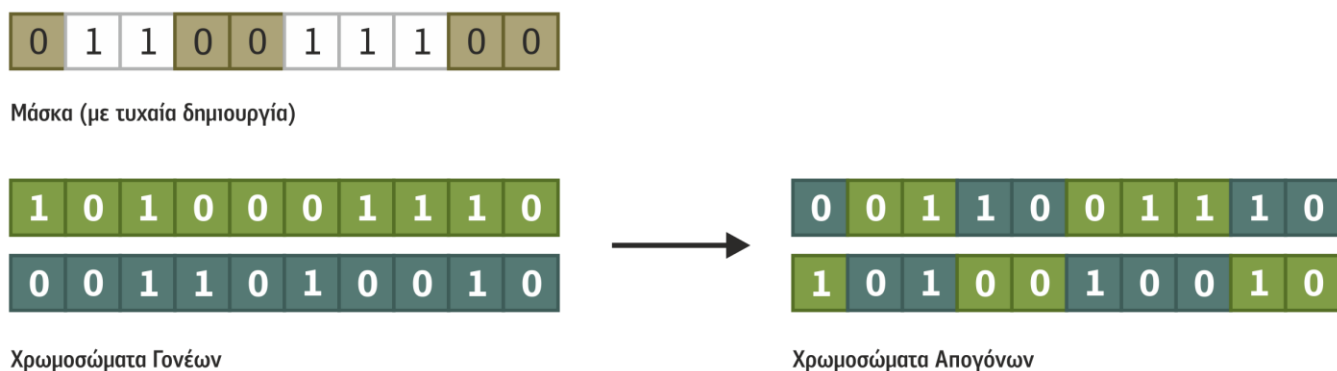
Ομοιόμορφη διασταύρωση



Σχήμα 4.41 Ομοιόμορφη τυχαία Διασταύρωση

Για την υλοποίηση αυτής της μεθόδου διασταύρωσης μπορούμε να χρησιμοποιήσουμε ως οδηγό μια τυχαία δημιουργημένη μάσκα σε μορφή μονοδιάστατου πίνακα με άσσους και μηδενικά, της οποίας οι μεν άσσοι ευνοούν τον πρώτο γονέα, ο οποίος δίνει την τιμή του γονιδίου του σε εκείνη τη θέση στον πρώτο απόγονο, τα δε μηδενικά ευνοούν το δεύτερο γονέα. Ο άλλος απόγονος παίρνει για κάθε γονιδιακή θέση το αλληλόμορφο του εκάστοτε μη ευνοημένου γονέα σε εκείνη τη θέση.

Ομοιόμορφη διασταύρωση με μάσκα



Σχήμα 4.42 Ομοιόμορφη Διασταύρωση με χρήση μάσκας

4.6.11 Μετάλλαξη (Mutation)

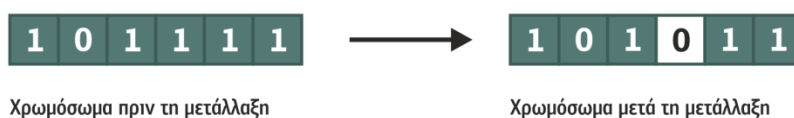
Στον φυσικό κόσμο, ο βιολογικός γονικός οργανισμός προωθεί αντίγραφα του γενετικού του κώδικα στους απογόνους του. Συνήθως όμως, κάποιο λάθος στην αντιγραφή μπορεί να προκαλέσει μια ωφέλιμη αλλαγή. Ο σκοπός της **μετάλλαξης** (mutation) είναι να εισαγάγει θόρυβο και, ειδικότερα, νέα αλληλόμορφα γονίδια εντός του πληθυσμού. Η μετάλλαξη είναι χρήσιμη, γιατί βοηθάει να βρεθούν τα τοπικά ακρότατα που δεν έχουν ανακαλυφθεί ακόμα, δεδομένου ότι βοηθά να εξερευνηθούν νέες περιοχές του πολυδιάστατου χώρου λύσεων. Εάν ο ρυθμός μετάλλαξης είναι πολύ υψηλός, μπορεί να οδηγήσει σε απώλεια καλών γονιδίων, με επακόλουθο να μειωθεί η εκμετάλλευση των περιοχών υψηλής καταλληλότητας του χώρου λύσεων. Για να

πραγματοποιηθεί η μετάλλαξη, επιλέγεται πρώτα το ποσοστό m που θα καθορίσει το πλήθος των μελών στα οποία θα εφαρμοστεί η μετάλλαξη. Το ποσοστό m συνήθως είναι πολύ μικρό ($< 10\%$) και πολλές φορές μηδενικό. Σε περίπτωση που το m είναι μηδενικό, δεν χρησιμοποιείται καθόλου ο τελεστής μετάλλαξης, με κίνδυνο ο ΓΑ να εγκλωβιστεί σε τοπικά ακρότατα ή να συγκλίνει πρόωρα.

Μόλις έχει επιλεγεί ένα γονίδιο για μετάλλαξη, η ίδια η μετάλλαξη μπορεί να λάβει διάφορες μορφές. Αυτό, πάλι, εξαρτάται από την υλοποίηση του γενετικού αλγόριθμου. Η μετάλλαξη στους ΓΑ υλοποιείται γενικά μέσω της τυχαίας αλλαγής κάποιων εκ των γονιδίων του χρωμοσώματος. Οι αλλαγές που επιφέρονται στα σχήματα από τη μετάλλαξη δεν έχουν σχέση με την απόδοσή τους.

Στην περίπτωση της κωδικοποίησης με δυαδικό αλφάβητο (βλέπε Σχήμα 4.43), η απλή μετάλλαξη ενός μόνο γονιδίου οδηγεί στο συμπλήρωμα της τιμής αυτού του γονιδίου, δηλαδή το ένα 1 γίνεται 0 και το αντίστροφο.

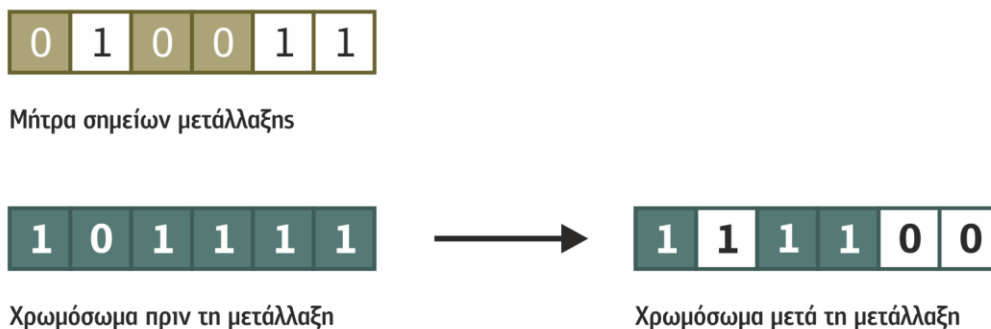
Μετάλλαξη ενός σημείου



Σχήμα 4.43 Η διαδικασία της μετάλλαξης σε δυαδικές συμβολοσειρές

Συνήθως, το σημείο ή τα σημεία μετάλλαξης επιλέγονται τυχαία ή με τη βοήθεια μιας μήτρας σημείων μετάλλαξης (βλέπε Σχήμα 4.44).

Μετάλλαξη τριών σημείων βάσει μήτρας



Σχήμα 4.44 Η διαδικασία της μετάλλαξης 3 σημείων βάσει μήτρας

Στην περίπτωση της μη δυαδικής κωδικοποίησης του χρωμοσώματος, απαιτούνται πιο περίπλοκες μέθοδοι μετάλλαξης. Η πιο απλή από αυτές είναι η μετάλλαξη τυχαίας ανταλλαγής, όπου γίνεται τυχαία επιλογή δύο γονιδίων και ανταλλαγή των θέσεών τους μέσα στο ίδιο χρωμόσωμα, όπως αυτή φαίνεται στο σχήμα 4.45. Άλλη είναι η **μετάλλαξη scrabble lists** (scrabble lists mutation) κατά την οποία αποκόπτεται με τυχαία επιλογή μια υπολίστα γονιδίων μέσα στο χρωμόσωμα, τα οποία κατά τη μετάλλαξη αναδιατάσσονται επίσης με τυχαίο τρόπο. Τέλος, στην **ομοιόμορφη μετάλλαξη** (uniform mutation), επιλέγονται εκτός του χρωμοσώματος που θα υποστεί μετάλλαξη και ένα δεύτερο χρωμόσωμα που θα οδηγήσει τη μετάλλαξη. Από το πρώτο χρωμόσωμα κρατούνται τυχαία κάποια γονίδια στις θέσεις τους (έστω το 50%) και τα υπόλοιπα ταξινομούνται βάσει της θέσης τους στο δεύτερο χρωμόσωμα. Στο παράδειγμα του σχήματος 4.45, από το x1 κρατούνται τα 2,3,5,6 και τα υπόλοιπα ταξινομούνται βάσει της θέσης τους στο άλλο χρωμόσωμα, δηλαδή

τοποθετούνται μέσα στις κενές θέσεις στο χρωμόσωμα με τη σειρά 8,4,7,1. Η ομοιόμορφη μετάλλαξη μπορεί να γίνει και σε ζεύγη χρωμοσωμάτων, κρατώντας κάποια γονίδια και από τα δυο χρωμοσώματα στη θέση τους και ταξινομώντας τα υπόλοιπα με βάση τη θέση τους στο άλλο χρωμόσωμα.

<p>Τυχαία ανταλλαγή</p> <p>Πριν από την μετάλλαξη x_1 1 2 3 4 5 6 7 8</p> <p>Μετά την μετάλλαξη x_1 1 5 3 4 2 6 7 8</p>	<p>Scrabble sublists</p> <p>Πριν από την μετάλλαξη x_1 1 2 3 4 5 6 7 8 τυχαία επιλογή αποκοπής τυχαία σειρά 4 2 3 5</p> <p>Μετά την μετάλλαξη x_1 1 4 2 3 5 6 7 8</p>
<p>Ομοιόμορφη μετάλλαξη ενός χρωμοσώματος</p> <p>Πριν από την μετάλλαξη x_1 1 2 3 4 5 6 7 8 x_2 8 6 4 2 7 5 3 1</p> <p>Μετά την μετάλλαξη x_1 8 2 3 4 5 6 7 1</p>	<p>Ομοιόμορφη μετάλλαξη ζεύγους χρωμοσώματος</p> <p>Πριν από την μετάλλαξη x_1 1 2 3 4 5 6 7 8 x_2 8 6 4 2 7 5 3 1</p> <p>Μετά την μετάλλαξη x_1 8 2 3 4 5 6 7 1 x_2 8 4 5 2 6 7 3 1</p>

Σχήμα 4.45 Τεχνικές μετάλλαξης σε μη δυαδικές συμβολοσειρές

Σε πιο σύνθετους τύπους δεδομένων, μια τιμή μπορεί να επιλεγεί τυχαία από μια συλλογή με πιθανές τιμές. Σε γενετικές αναπαραστάσεις με υψηλότερο βαθμό αποκωδικοποίησης, η μετάλλαξη γίνεται όλο και πιο περίπλοκη. Σε κάθε περίπτωση, το μόνο που απαιτείται είναι να είναι η μέθοδος της μετάλλαξης αρκετά γενική, ώστε να μπορεί να προκαλέσει την εμφάνιση οποιουδήποτε πιθανού αλληλόμορφου εντός του πληθυσμού.

4.6.12 Γενεαλογική Αντικατάσταση

Μετά την εφαρμογή των γενετικών τελεστών, ακολουθεί η οριστικοποίηση της νέας γενεάς με την αντικατάσταση των γονέων από τους απόγονους τους, η οποία ονομάζεται **γενεαλογική αντικατάσταση** (generational replacement). Αξίζει να αναφέρουμε τις 3 κατηγορίες αντικατάστασης που έχουν επικρατήσει:

- **Ολική αντικατάσταση** (total replacement): ολόκληρος ο πληθυσμός των γονέων αντικαθίσταται από τους απόγονους.
- **Μερική αντικατάσταση** (partial replacement): ένα μέρος του νέου πληθυσμού των γονέων (συνήθως αυτοί με τη χαμηλότερη απόδοση) αντικαθίσταται από ένα ισοδύναμο πλήθος από απογόνους (συνήθως αυτούς με την υψηλότερη απόδοση).

- **Αντικατάσταση σταθερής κατάστασης** (steady state replacement): κάθε απόγονος που παράγεται αντικαθιστά έναν γονέα (συνήθως τον χειρότερο) και ταυτόχρονα μπορεί να πάρει μέρος στην παραγωγή του επόμενου απόγονου.

Υπάρχουν δύο ανταγωνιστικοί παράγοντες που πρέπει να εξισορροπούνται κατά τη διαδικασία αντικατάστασης, όταν αυτή δεν είναι ολική: η *επιλεκτική πίεση* και η *γενετική ποικιλομορφία*.

Επιλεκτική πίεση (selective pressure) είναι η τάση να επιλέγονται μόνο τα καλύτερα μέλη της τρέχουσας γενεάς, για να διαδοθούν στην επόμενη. Είναι απολύτως απαραίτητη, για να κατευθύνει τον γενετικό αλγόριθμο σε ένα βέλτιστο.

Γενετική ποικιλομορφία (generational diversity) είναι η διατήρηση ενός ποικιλόμορφου πληθυσμού λύσεων. Είναι επίσης απαραίτητη, για να εξασφαλιστεί ότι ο χώρος λύσεων έχει διερευνηθεί επαρκώς, ιδιαίτερα στα αρχικά στάδια της εκτέλεσης του αλγόριθμου.

Η υπερβολική επιλεκτική πίεση μπορεί να μειώσει τη γενετική ποικιλομορφία, ώστε το ολικό βέλτιστο να παραβλέπεται και ο γενετικός αλγόριθμος να συγκλίνει σε ένα κάποιο τοπικό βέλτιστο. Ωστόσο, με πολύ μικρή επιλεκτική πίεση ο αλγόριθμος δεν μπορεί να συγκλίνει σε μια βέλτιστη λύση σε εύλογο χρονικό διάστημα. Μια σωστή ισορροπία ανάμεσα στην επιλεκτική πίεση και τη γενετική ποικιλομορφία είναι απαραίτητη, για να συγκλίνει ο αλγόριθμος στο ολικό βέλτιστο σε ένα εύλογο χρονικό διάστημα.

4.6.13 Εκλεκτικές Στρατηγικές

Μια τεχνική που χρησιμοποιούμε, πριν οριστικοποιηθεί η κάθε γενεά, είναι ο **Ελιτισμός** (Elitism). Βασική αρχή του Ελιτισμού είναι η εύρεση του ατόμου με την υψηλότερη τιμή απόδοσης από την παλαιά γενεά και η αντιγραφή του στη νέα γενεά, ακόμα και αν οι εφαρμοζόμενες τεχνικές αναπαραγωγής απαιτήσουν την εξάλειψή του. Αυτό είναι μία εξασφάλιση ότι η καλύτερη λύση δεν θα χαθεί κατά τη γονική επιλογή ή κατά την εφαρμογή των γενετικών τελεστών και τελικά θα είναι η λύση στην οποία πρέπει να συγκλίνει όλος ο πληθυσμός.

Η αντιγραφή του τρέχοντος καλύτερου χρωμοσώματος στην επόμενη γενεά γίνεται με δύο διαφορετικούς τρόπους:

- Μόνο εάν δεν υπάρχει ήδη
- Ούτως ή άλλως

Υπάρχουν διάφορες εκδοχές για τη θέση μέσα στον πληθυσμό που θα καταλάβει:

- θέση ενός τυχαία επιλεγμένου χρωμοσώματος,
- θέση του χρωμοσώματος με τη μικρότερη απόδοση,
- χωρίς αντικατάσταση κάποιου άλλου χρωμοσώματος, οπότε προκαλείται μικρή σταδιακή αύξηση του μεγέθους του πληθυσμού.

4.6.14 Ολοκλήρωση ΓΑ

Το απλούστερο κριτήριο για να ολοκληρωθεί ένας ΓΑ είναι να βρεθεί στον πληθυσμό μιας γενεάς του μια λύση που να έχει τιμή απόδοσης πάνω από μια προκαθορισμένη ικανοποιητική τιμή, αν κάτι τέτοιο είναι εύκολο από το πρόβλημα να καθοριστεί.

Άλλο κριτήριο είναι η σύγκλιση του ΓΑ. Ένας ΓΑ συγκλίνει, όταν η συγκέντρωση των μελών του πληθυσμού βρίσκεται σε μία στενή περιοχή του συνολικού χώρου των λύσεων προσεγγίζοντας ένα τοπικό ή γενικό βέλτιστο. Αυτό φαίνεται, όταν οι γονότυποι των μελών του πληθυσμού είναι σχεδόν όλοι ίδιοι και ο

ανασυνδυασμός τους δεν μπορεί να παράγει νέες λύσεις, με αποτέλεσμα τη στασιμότητα στην εξέλιξη του αλγόριθμου. Η μόνη πιθανότητα παραγωγής νέας βελτιωμένης λύσης σε μια τέτοια περίπτωση είναι μέσω της μετάλλαξης, αν και έχει μικρή πιθανότητα επιτυχίας.

Η σύγκλιση του ΓΑ μπορεί να αποτελέσει ένα κριτήριο τερματισμού του. Όμως, δεν είναι εύκολο να εντοπιστεί και για το λόγο αυτόν χρησιμοποιείται ως εναλλακτικό κριτήριο τερματισμού η ολοκλήρωση ενός μέγιστου αριθμού γενεών που έχει καθοριστεί από την έναρξη του ΓΑ.

Έτσι, ένας ΓΑ μπορεί να τερματιστεί:

- σε περίπτωση λύσης με ικανοποιητική τιμή απόδοσης,
- όταν επί μεγάλου αριθμού συνεχόμενων γενεών δεν παράγεται καλύτερη ή διαφορετική λύση,
- εάν το πλήθος των γενεών φτάσει σε έναν προκαθορισμένο αριθμό, π.χ. 500 γενεές.

4.6.15 Διαφορές από άλλες τεχνικές

Οι γενετικοί αλγόριθμοι διαφέρουν από άλλες τεχνικές αναζήτησης στα εξής σημεία:

- Εξισορροπούν την αναζήτηση νέων σημείων στο χώρο αναζήτησης με την εκμετάλλευση των ήδη γνωστών πληροφοριών.
- Είναι τυχαίοι αλγόριθμοι, δεδομένου ότι χρησιμοποιούν τελεστές των οποίων τα αποτελέσματα εξαρτώνται από τις πιθανότητες, με την έννοια ότι είναι βασισμένα στην τιμή ενός τυχαίου αριθμού. Δηλαδή, οι ΓΑ χρησιμοποιούν πιθανολογικούς κανόνες μετάβασης, όχι αιτιοκρατικούς.
- Λειτουργούν σε διάφορες λύσεις ταυτόχρονα, συγκεντρώνοντας πληροφορίες από τα τρέχοντα σημεία αναζήτησης για το αμέσως επόμενο επαναληπτικό βήμα. Η δυνατότητά τους να διατηρούν πολλαπλές λύσεις ταυτόχρονα τους καθιστά λιγότερο ευαίσθητους στο πρόβλημα των τοπικών μεγίστων και του θορύβου.
- Λειτουργούν με μια κωδικοποίηση του συνόλου των παραμέτρων, όχι με τις ίδιες τις παραμέτρους.
- Εκτελούν την αναζήτηση σε έναν πληθυσμό σημείων, όχι σε ένα μοναδικό σημείο.
- Χρησιμοποιούν τη συνάρτηση καταλληλότητας, για να στοχεύσουν σημεία στο χώρο λύσεων χωρίς άλλη βοηθητική γνώση.

4.6.16 Εφαρμογές Γενετικών Αλγορίθμων

Οι γενετικοί αλγόριθμοι έχουν εφαρμογή σε πολύπλοκα προβλήματα, κυρίως βελτιστοποίησης και χρονοδρομολόγησης. Προβλήματα με τέτοια χαρακτηριστικά που λύνονται πολύ αποτελεσματικά με την εφαρμογή ΓΑ είναι:

- εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων,
- επεξεργασία εικόνων,
- συνδυαστική βελτιστοποίηση (*καταμερισμός εργασιών, ωρολόγιο πρόγραμμα, πλανόδιος πωλητής κτλ*),
- σχεδίαση (*λύση+βελτιστοποίηση+νέες ιδέες κατασκευής*),
- ανακάλυψη κανόνων

Ειδική κατηγορία εφαρμογής ΓΑ είναι ο Γενετικός Προγραμματισμός (Genetic Programming), που έχει ως στόχο την αυτόματη κατασκευή κώδικα προγραμμάτων Η/Υ από τυπικές αλγοριθμικές δομές (IF-THEN-ELSE, WHILE κτλ.) που παίζουν το ρόλο των γονιδίων ενός χρωμοσώματος που αντιστοιχεί στον κώδικα που ζητείται να κατασκευαστεί.

Βιβλιογραφία/Αναφορές

- Aamodt, A., & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues - Methodological Variations, and System Approaches. *Artificial Intelligence Communications-IOS Press*, 7(1), 39-59. Ανακτήθηκε από <http://ibug.doc.ic.ac.uk/media/uploads/documents/courses/CBR-AamodtPlaza.pdf>
- Beale, R., & Jackson, T. (1990). *Neural Computing: An Introduction*. N.Y.: Adam Hilger.
- Colin, A. (1996). Building Decision Trees with the ID3 Algorithm. *Dr Dobb's Journal of Software Tools for Professional Programmer*, 21(6), 107-109.
- De Jong, K. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems* (Διδακτορική Διατριβή, Πανεπιστήμιο Michigan, ΗΠΑ). Ανακτήθηκε από <http://deepblue.lib.umich.edu/handle/2027.42/4507>
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Wokingham, ΗΠΑ: Addison-Wesley.
- Qu, R. (2002). *Case-Based Reasoning for Course Timetabling Problems* (Διδακτορική Διατριβή Πανεπιστήμιο Nottingham). Ανακτήθηκε από <http://www.cs.nott.ac.uk/~rxq/files/PhDThesis.pdf>
- Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning*. ΗΠΑ: Morgan Kaufmann.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems* (2η έκδοση). Καίμπριτζ, MA: MIT Press. (1η έκδοση 1975 University of Michigan Press)
- Leake, D. B. (1996). *Case-Based Reasoning: Experiences, Lessons and Future Directions*. Menlon Park, CA: AAAI/MIT Press
- Maher, M., Balachandran, B., & Meizhang, D. (1995). *CBR in Design*. New Jersey: Lawrence Erlbaum Publishers.
- Mitchell, T.M. (1997). *Machine Learning*. Maidenhead, H.B.: McGraw-Hill International Editions.
- Nordlund, J., & Schafer, H. (2006). *Case-Based Reasoning in a Support system* (Master in CS, Πανεπιστήμιο Umea). Ανακτήθηκε από <http://www8.cs.umu.se/education/examina/Rapporter/NordlundSchafer.pdf>
- Pao, Yoh-Han (1989). *Adaptive Pattern Recognition and Neural Networks*. Wokingham, ΗΠΑ: Addison-Wesley.
- Russel, S. J. , & Norvig, P. (2003). *Artificial Intelligence-A Modern Approach* (2^η έκδοση). Upper Saddle River, Νιου Τζέρσεϋ: Pearson Prentice Hall (1^η έκδοση 1995).
- Utgoff, P. E. (1989). Incremental Induction of Decision Trees. *Machine Learning*, 4, 161-186. Ανακτήθηκε από <http://people.cs.umass.edu/~utgoff/papers/mlj-id5r.pdf>
- Whitley, D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing*, 4, 65–85. Ανακτήθηκε από <http://www.cs.colostate.edu/~genitor/MiscPubs/tutorial.pdf>
- Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κοκκοράς, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστημίου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Ποια από τα παρακάτω συστήματα ανήκουν στο χώρο της μηχανικής μάθησης;

1. Συστήματα Κανόνων,
2. Δένδρα Απόφασης,
3. Έμπειρα Συστήματα,
4. Νευρωνικά Δίκτυα.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 2η και η 4η.

Κριτήριο αξιολόγησης 2

Ποιο είναι το πρώτο βήμα για τη δημιουργία ενός δένδρου παραγωγής;

1. Συγκέντρωση υποδειγματικών περιπτώσεων,
2. Δημιουργία συνόλου βασικών χαρακτηριστικών του δεδομένου προβλήματος,
3. Επιλογή χαρακτηριστικού, για να αποτελέσει τη ρίζα του δένδρου παραγωγής.

Απάντηση/Λύση

Η σωστή απάντηση είναι η 2^η.

Κριτήριο αξιολόγησης 3

Τι από τα παρακάτω ισχύει για ένα τεχνητό νευρωνικό δίκτυο;

1. Δεν διαχειρίζεται σύμβολα.
2. Μαθαίνει από παραδείγματα.
3. Αφού εκπαιδευτεί, συμπεριφέρεται σωστά, ακόμα και αν καταστραφούν κάποιοι εσωτερικοί του νευρώνες.
4. Αφού εκπαιδευτεί σωστά, αναγνωρίζει στη συνέχεια ήδη τις γνωστές περιπτώσεις αλλά και νέες άγνωστες σε αυτό.

Απάντηση/Λύση

Όλες οι απαντήσεις είναι σωστές

Κριτήριο αξιολόγησης 4

Ένα τεχνητό νευρωνικό δίκτυο:

1. Εκπαιδεύεται με ένα σύνολο γνωστών περιπτώσεων, για να μάθει να αναγνωρίζει νέες περιπτώσεις.
2. Εκπαιδεύεται με ένα σύνολο γνωστών περιπτώσεων, για να μάθει να βρίσκει λύση σε ένα πρόβλημα.
3. Εκπαιδεύεται με ένα σύνολο γνωστών περιπτώσεων, για να μάθει να αναγνωρίζει μία από αυτές, όταν ξαναπαρουσιαστεί.
4. Εκπαιδεύεται με ένα σύνολο γνωστών περιπτώσεων, για να μάθει να απορρίπτει άγνωστες περιπτώσεις.

Απάντηση/Λύση

Σωστή απάντηση είναι μόνο η 1^η.

Κριτήριο αξιολόγησης 5

Ποια από τα παρακάτω απαιτούνται να προσδιοριστούν κατά τη δημιουργία ενός Γενετικού Αλγορίθμου;

1. Αναπαράσταση γνώσης,
2. Κωδικοποίηση χρωμοσώματος,
3. Χαρακτηριστικά διαχωρισμού,
4. Συνάρτηση καταλληλότητας,
5. Χαρακτηριστικά αναπαραγωγής πληθυσμού.

Απάντηση/Λύση

Σωστές απαντήσεις είναι οι 2^η, 4^η και 5^η

Κριτήριο αξιολόγησης 6

Έστω ότι θέλουμε να διασταυρώσουμε 2 γονείς με χρωμοσώματα $X1=001111000$ και $X2=100001111$ με διασταύρωση ενός σημείου που είναι το σημείο 4. Ποια θα είναι τα αποτελέσματα μετά τη διασταύρωση;

1. $X1'=001011000$ $X2'=100101111$
2. $X1'=100011000$ $X2'=001101111$
3. $X1'=001101111$ $X2'=100011000$
4. $X1'=001001111$ $X2'=100110000$

Απάντηση/Λύση

Η σωστή απάντηση είναι η 3^η.

Κριτήριο αξιολόγησης 7

Ο πιο πιθανός λόγος, για να ολοκληρώσει τη λειτουργία του ένας γενετικός αλγόριθμος είναι,όταν:

1. Ένα μέλος του πληθυσμού έχει προσεγγίσει την άριστη τιμή καταλληλότητας.
2. Ένα μέλος του πληθυσμού συμπίπτει με τη λύση που έχει δοθεί.
3. Δεν δημιουργούνται πλέον νέοι απόγονοι.
4. Έχει συμπληρωθεί ο προκαθορισμένος αριθμός γενεών.

Απάντηση/Λύση

Σωστές απαντήσεις είναι οι 1^η και 4^η.

ΚΕΦΑΛΑΙΟ 5 - Έμπειρα Συστήματα

Σύνοψη

Στο κεφάλαιο αυτό παρουσιάζεται η μηχανική συστημάτων βασισμένων στη γνώση με βασικό εκπρόσωπό τους τα Έμπειρα Συστήματα Βασισμένα σε Κανόνες (*Rule-Based Expert Systems*). Παρουσιάζονται η δομή και η λειτουργία τους, η συγκέντρωση και μηχανική της γνώσης, καθώς και περιπτώσεις εφαρμογών τους. Ιδιαίτερη περίπτωση είναι η μηχανική της αβέβαιης και ασαφούς γνώσης που παρουσιάζεται στο τέλος του κεφαλαίου με μικρά παραδείγματα.

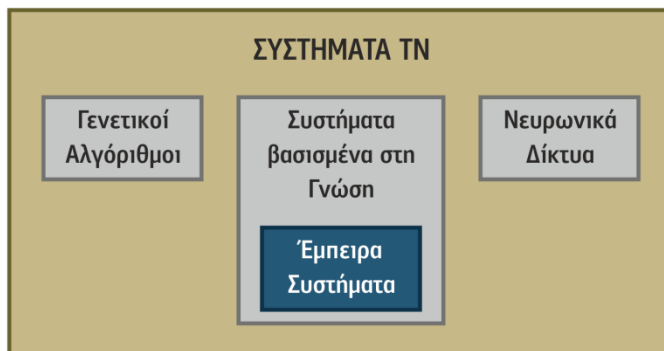
Προαπαιτούμενη γνώση

Αναπαράσταση γνώσης, Συστήματα Κανόνων.

5.1 Γενικά

Ένα Έμπειρο Σύστημα-ΕΣ (*Expert System - ES*) είναι εξ ορισμού ένα σύστημα το οποίο προσπαθεί να επιδείξει ικανότητες στο να λαμβάνει αποφάσεις παρόμοιες με αυτές ενός ειδήμονα (εμπειρογνώμονα) σε ένα γνωστικό τομέα. Πιο απλά, ένα ΕΣ είναι διαλογικό μηχανογραφικό εργαλείο σχεδιασμένο να λύνει δύσκολα προβλήματα λήψης αποφάσεων τα οποία βασίζονται σε γνώση συγκεντρωμένη από ειδήμονες. Υπό αυτό το πρίσμα, ένα ΕΣ αναμένεται να ενεργεί σε όλα του τα σημεία παρόμοια με τον τρόπο με τον οποίο θα ενεργούσε ένας ειδήμονας.

Τα ΕΣ είναι κλάδος της ΤΝ που υπάγεται στο γενικότερο σχήμα των Συστημάτων Βασισμένων στη Γνώση-ΣΒΓ (*Knowledge Based Systems - KBS*). Η διαφορά των ΕΣ από τα ΣΒΓ είναι ότι τα πρώτα περιέχουν ειδική γνώση, ενώ τα δεύτερα γενική γνώση ενός συγκεκριμένου κόσμου.



Σχήμα 5.1 Η θέση των Εμπειρών Συστημάτων στο χώρο της Τεχνητής Νοημοσύνης

Τα ΕΣ μπορούν να διαχωριστούν από τα άλλα συστήματα του χώρου στο ότι:

- Εξετάζουν προβλήματα ρεαλιστικής περιπλοκότητας τα οποία απαιτούν να αξιοποιηθεί σεβαστό απόθεμα ανθρώπινης εμπειρίας.
- Πρέπει να είναι ικανά για αξιολόγηση, επεξήγηση και δικαιολόγηση λύσεων, καθώς και παροχή συμβουλών, για να πειστεί ο χρήστης.
- Πρέπει να είναι γρήγορα και αξιόπιστα, ώστε να είναι χρήσιμα ως εργαλεία.

Για να σχεδιαστεί ένα ΕΣ, δύο είναι τα βασικά βήματα: η απόκτηση της απαραίτητης γνώσης για το προς επίλυση πρόβλημα και η αναπαράστασή της (βλέπε [Κεφάλαιο 2](#)).

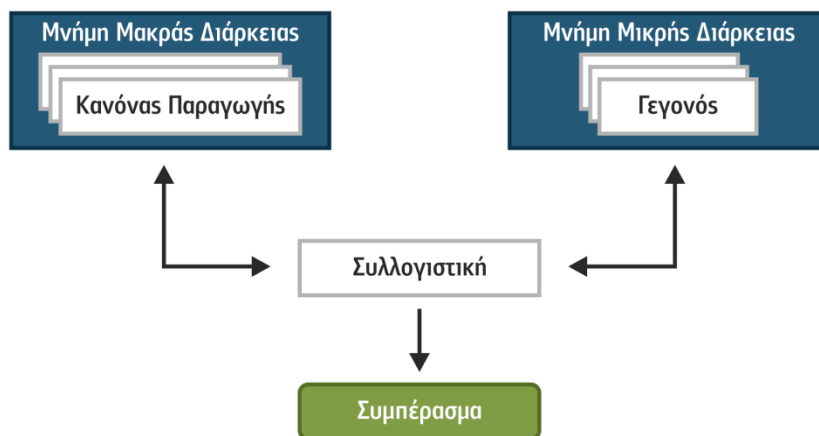
Απόκτηση γνώσης(knowledge acquisition) καλείται η εξειδικευμένη γνώση που πρέπει να προσκτηθεί σε σχέση με:

- τα γεγονότα που περιγράφουν τον κόσμο του προβλήματος,
- τις «εμπειρικές γνώσεις» που συσσωρεύει ένας ειδικός κατά την πάροδο των ετών που ασχολείται με το πρόβλημα.

5.2 Η Δομή ενός ΕΣ Βασισμένου σε Κανόνες

Η δομή των **Έμπειρων Συστημάτων Βασισμένων σε Κανόνες-ΕΣΒΚ** (Rule-Based Expert Systems-RBES) βασίζεται στο **μοντέλο συστήματος παραγωγής** (production system model), το οποίο πρότειναν οι γνωστοί επιστήμονες του χώρου της TN Newell και Simon από το Πανεπιστήμιο Carnegie-Mellon στις αρχές της δεκαετίας του '70.

Το μοντέλο συστήματος παραγωγής βασίζεται στην ιδέα ότι οι άνθρωποι λύνουν προβλήματα εφαρμόζοντας τη γνώση τους σε δεδομένο πρόβλημα που παρουσιάζεται μέσω πληροφοριών που αφορούν τα δεδομένα του· η γνώση αποθηκεύεται στη **μνήμη μακράς διάρκειας** (long-term memory) και οι πληροφορίες στη **μνήμη μικρής διάρκειας** (short-term memory) (βλέπε Σχήμα 5.2).

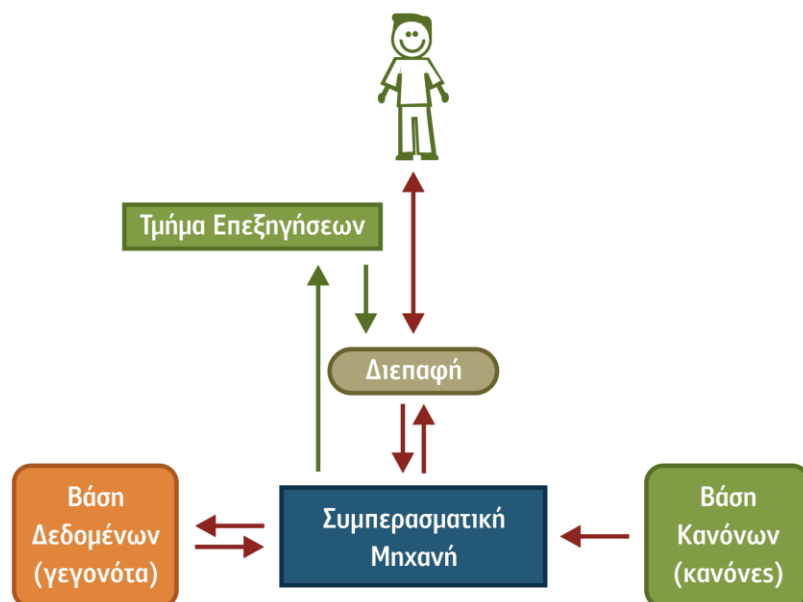


Σχήμα 5.2 Μοντέλο συστήματος παραγωγής

Μια τυπική δομή ενός ΕΣ, επιτελείται από:

- τη **Βάση Δεδομένων** (DataBase) ή **Μνήμη Εργασίας** (Working Memory) στο ρόλο της μνήμης μικρής διάρκειας,
- τη **Βάση Κανόνων** (RuleBase) ή **Βάση Γνώσης** (KnowledgeBase) ή **Παραγωγική Μνήμη** (Productive Memory) στο ρόλο της μνήμης μακράς διάρκειας,
- Το **Μηχανισμό Εξαγωγής Συμπερασμάτων** (Inference Mechanism) ή **Συμπερασματική Μηχανή** (Inference Engine) στο ρόλο της Συλλογιστικής.
- Το **Τμήμα Διεπαφής** (Interface) που επιτρέπει την επικοινωνία μεταξύ του χρήστη και του ΕΣ.

Επιπλέον χαρακτηριστικό στοιχείο της δομής είναι το **Τμήμα Επεξηγήσεων** (Explanations Facilities).



Σχήμα 5.3 Τυπική δομή ενός ΕΣ

Στη Βάση Κανόνων, το πεδίο γνώσης που είναι χρήσιμο για την επίλυση του προβλήματος παρουσιάζεται ως ένα σύνολο κανόνων. Κάθε κανόνας ορίζει μια σχέση, σύσταση ή οδηγία. Όταν πληρούνται οι υποθέσεις του μέρους ΑΝ ενός κανόνα, ο κανόνας αυτός τοποθετείται από τη δομή ελέγχου στο σύνολο σύγκρουσης και, αν κατά την επίλυση συγκρούσεων επιλεγεί ο συγκεκριμένος κανόνας, τότε ενεργοποιείται και εκτελούνται οι ενέργειες του μέρους ΤΟΤΕ εκτελούνται (βλέπε [κεφάλαιο 3](#)).

Η Βάση Δεδομένων περιλαμβάνει σε κάθε στιγμιότυπο εκτέλεσης ένα σύνολο γεγονότων που ισχύουν το συγκεκριμένο στιγμιότυπο. Η δομή ελέγχου συγκρίνει τα γεγονότα με τις υποθέσεις του μέρους ΑΝ των κανόνων κατά τη σύγκριση προτύπων που πραγματοποιεί στην αρχή κάθε κύκλου εκτέλεσης του ΕΣ.

Η Συμπερασματική Μηχανή πραγματοποιεί τη συλλογιστική σύμφωνα με την οποία το ΕΣ φθάνει σε μια λύση. Ειδικότερα, χρησιμοποιείται από το σύστημα για να δικαιολογήσει ή να συναγάγει συμπεράσματα όπως θα ενεργούσε ένας ειδήμονας, δεδομένων παρόμοιων συνθηκών. Ο ρόλος της είναι να ξέρει πότε και πώς να εφαρμόζει την έως τώρα αποθηκευμένη γνώση με τη βοήθεια της δομής ελέγχου που διαθέτει, για να συνδέει τους κανόνες στη Βάση Κανόνων με τα γεγονότα στη Βάση Δεδομένων και να εκτελεί κανόνες με ικανοποιημένους ισχυρισμούς βάσει στρατηγικών και ευρετικών λειτουργιών που περιέχονται στους κανόνες ή δηλώνονται εξωτερικά πριν από την εκτέλεση του ΕΣ.

Η Τμήμα Διεπαφής είναι το μέσο επικοινωνίας μεταξύ των χρηστών που αναζητούν μια λύση στο πρόβλημα και ενός ΕΣ.

Το Τμήμα Επεξηγήσεων επιτρέπει στο χρήστη να ζητήσει από το ΕΣ να του παρουσιάσει μέσω της διεπαφής πώς κατέληξε σε ένα συγκεκριμένο συμπέρασμα και γιατί χρειάζεται ένα συγκεκριμένο γεγονός. Ένα ΕΣ πρέπει να είναι σε θέση να εξηγήσει το σκεπτικό του και να δικαιολογήσει τις συμβουλές, την ανάλυση ή το συμπέρασμά του με τη βοήθεια του ιστορικού που διατηρεί για την πυροδότηση των κανόνων.

5.3 Χαρακτηριστικά ενός ΕΣ

Ένα ΕΣ κατασκευάζεται, για να λειτουργεί σε ένα στενό, εξειδικευμένο τομέα. Έτσι, το πιο σημαντικό χαρακτηριστικό του είναι η υψηλής ποιότητας απόδοσή του. Είναι προβληματικό αν το σύστημα λύσει ένα πρόβλημα γρήγορα, αλλά ο χρήστης δεν μείνει ικανοποιημένος σε περίπτωση κατά την οποία το αποτέλεσμα είναι λάθος. Ωστόσο, η ταχύτητα της επίτευξης μιας λύσης είναι πολύ σημαντική. Ακόμη και η πιο ακριβής απόφαση ή διάγνωση δεν μπορεί να είναι χρήσιμη, αν, όταν βρεθεί, είναι πολύ αργά για να εφαρμοστεί.

Τα ΕΣ εφαρμόζουν ευρετικούς μηχανισμούς (heuristics), για να καθοδηγήσουν τη συλλογιστική που ακολουθούν και να μειώσουν την περιοχή αναζήτησης μιας λύσης. Διαθέτουν, όμως, και ένα μοναδικό χαρακτηριστικό, το οποίο είναι η ικανότητά τους να εξηγούν τις αποφάσεις τους.

Τα ΕΣ χρησιμοποιούν συμβολική λογική κατά την επίλυση ενός προβλήματος. Τα σύμβολα χρησιμοποιούνται για να αντιπροσωπεύουν διαφορετικούς τύπους γνώσης, όπως γεγονότα, έννοιες και κανόνες με βάση τη γνώση που έχουν δοθεί από τον ειδικό τομέα. Όμως, ακόμη και ο πιο ευφυής ειδήμων είναι μόνο ένας άνθρωπος και μπορεί να κάνει λάθη. Και τα ΕΣ που δομούνται από τη γνώση των ειδικών πρέπει, επίσης, να έχουν τη δυνατότητα να κάνουν λάθη. Πρέπει να λαμβάνουμε υπόψη μας ότι στα ΕΣ τα λάθη είναι πιθανά, αν και στις περισσότερες, τουλάχιστον, περιπτώσεις μπορούμε να βασιστούμε σε λύσεις τις οποίες παρέχουν, όπως και οι αποφάσεις των ειδικών μπορεί να είναι εσφαλμένες, ακόμη και αν έχουμε εμπιστοσύνη σε αυτούς.

Στα ΕΣ, η γνώση χωρίζεται από την επεξεργασία της (η επεξεργασία της Βάσης Γνώσης και της Συμπερασματικής Μηχανής είναι διαχωρισμένη). Όταν χρησιμοποιείται ένα κέλυφος εμπειρου συστήματος (expert system shell), ένας μηχανικός γνώσης ή ένας εμπειρογνώμονας εισάγει απλώς τους κανόνες στη βάση γνώσης. Κάθε νέος κανόνας προσθέτει κάποια νέα γνώση και κάνει το ΕΣ πιο νοήμον. Αντίθετα, ένα συμβατικό πρόγραμμα είναι μείγμα της γνώσης και της δομής ελέγχου για την επεξεργασία αυτής της γνώσης. Η ανάμειξη προκαλεί δυσκολίες στην κατανόηση και την αναθεώρηση του κώδικα του προγράμματος, καθώς οποιαδήποτε αλλαγή στον κώδικα επηρεάζει τόσο τη γνώση όσο και την επεξεργασία της.

Η παραπάνω διαφορά μεταξύ ενός ΕΣ και ενός συμβατικού προγράμματος δεν είναι η μοναδική. Γενικά, οι διαφορές πηγάζουν από το ότι ένα ΕΣ αποτελείται από το τμήμα μηχανικής της γνώσης και τη Συμπερασματική Μηχανή, ενώ ένα συμβατικό πρόγραμμα αποτελείται από την κωδικοποίηση των δεδομένων του και τον αλγόριθμό του:

Έμπειρο Σύστημα = Γνώση + Συμπερασματική Μηχανή
Πρόγραμμα = Δεδομένα + Αλγόριθμος

Στον παρακάτω πίνακα 5.1 αποτυπώνονται οι βασικότερες διαφορές μεταξύ των δυο τύπων λογισμικού.

Διαφορές μεταξύ ΕΣ και συμβατικών προγραμμάτων	
Έμπειρα Συστήματα	Συμβατικά Προγράμματα
Προσομοίωση τρόπου επίλυσης προβλήματος	Προσομοίωση του ίδιου του προβλήματος
Γνώση σε επίπεδο συμβόλων	Δεδομένα σε επίπεδο υπολογισμών
Χρήση ευρετικών μεθόδων	Χρήση αλγορίθμων
Χειρισμός αβέβαιης και ασαφούς γνώσης	Δυσχέρεια στη χρήση μη-πλήρους γνώσης
Δυνατότητα μη μονοτονικής συλλογιστικής	Μη δυνατότητα χρήσης μη-μονοτονικής συλλογιστικής (δηλαδή συλλογιστική που αποτυπώνεται σε κανόνες με αναιρέσιμα συμπεράσματα)
Επεξήγηση δρόμου συλλογισμού	Ανυπαρξία αυτόματης επεξήγησης
Ευκολία στην κατανόηση και αναθεώρηση της αποτυπωμένης γνώσης	Δυσκολίες στην κατανόηση και την αναθεώρηση του κώδικα του προγράμματος

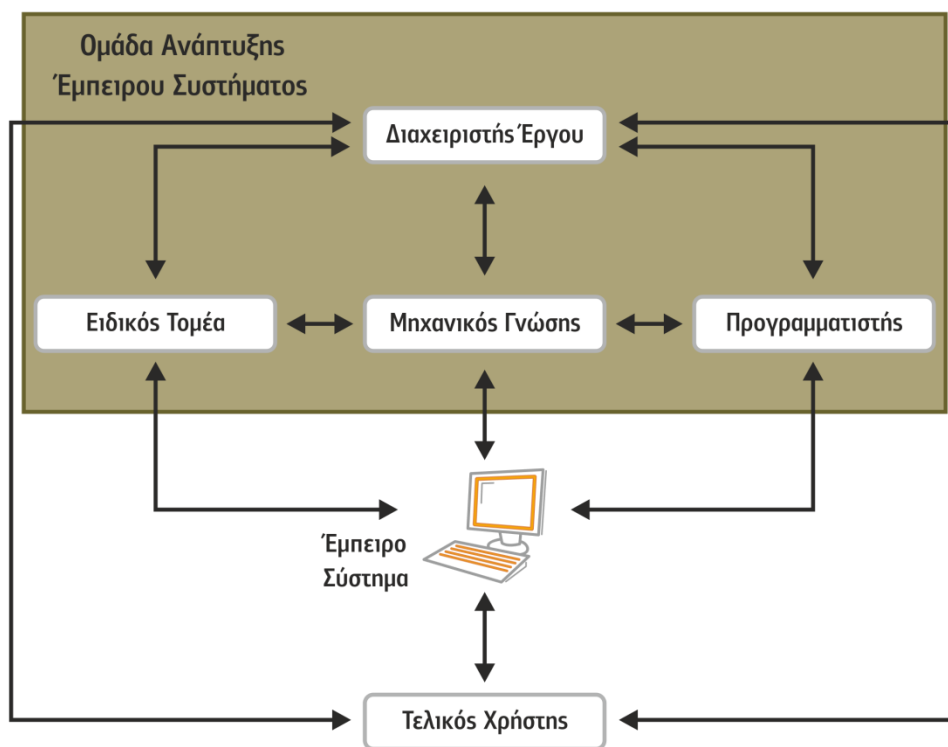
Πίνακας 5.1 Βασικές διαφορές μεταξύ ΕΣ και συμβατικών προγραμμάτων

5.4 Ανάπτυξη ενός ΕΣ

Υπάρχουν 5 κατηγορίες συμμετεχόντων σε μια ομάδα ανάπτυξης ενός ΕΣ:

- Ο ειδικός του τομέα ενδιαφέροντος,
- Ο μηχανικός γνώσης,
- Ο προγραμματιστής,
- Ο διαχειριστής του έργου,
- Ο τελικός χρήστης.

Η επιτυχία του υπό δημιουργία ΕΣ εξαρτάται ολοκληρωτικά από το πόσο καλά συνεργάζονται τα μέλη της ομάδας ανάπτυξης. Στο σχήμα 5.4 παρουσιάζονται οι συμμετέχοντες στην ομάδα ανάπτυξης του ΕΣ.



Σχήμα 5.4 Οι συμμετέχοντες στην ομάδα ανάπτυξης ενός ΕΣ

Ο **ειδικός τομέα** (domain expert) είναι ένα πρόσωπο με γνώσεις και δεξιότητες ικανό να λύνει προβλήματα σε μια συγκεκριμένη περιοχή ή τομέα. Στο δεδομένο τομέα πρέπει να διαθέτει τη μεγαλύτερη εμπειρία, η οποία πρέπει να αποτυπωθεί στο ΕΣ. Ως εκ τούτου, ο ειδικός πρέπει να είναι ικανός:

- να επικοινωνεί τη γνώση του/της,
- να είναι πρόθυμος/η να συμμετέχει στην ομάδα ανάπτυξης και
- να διαθέτει τον απαιτούμενο χρόνο για το σκοπό αυτό.

Ο ειδικός τομέα είναι το πιο σημαντικό πρόσωπο της ομάδας ανάπτυξης.

Ο **μηχανικός γνώσης** (knowledge engineer) είναι αυτός που θα σχεδιάσει, θα κατασκευάσει και θα ελέγξει το ΕΣ. Συνεργάζεται με τον ειδικό τομέα, για να μάθει πώς λύνεται το συγκεκριμένο πρόβλημα. Αναγνωρίζει ποια συλλογιστική χρησιμοποιεί ο ειδικός, για να χειριστεί τα γεγονότα και τους κανόνες, και

αποφασίζει πώς θα τους αναπαραστήσει στο ΕΣ. Επιλέγει αν θα προχωρήσει σε ανάπτυξη λογισμικού ή θα χρησιμοποιήσει ένα κέλυφος ΕΣ, αναζητεί κατάλληλες γλώσσες προγραμματισμού για την κωδικοποίηση της γνώσης. Είναι υπεύθυνος/η για τον έλεγχο, την αναθεώρηση και την ενσωμάτωση του ΕΣ συστήματος στο χώρο εργασίας.

Ο προγραμματιστής (programmer) είναι υπεύθυνος για τον προγραμματισμό του ΕΣ, περιγράφοντας τις γνώσεις του τομέα με τρόπο κατανοητό από έναν υπολογιστή. Εκτός της γνώσης συμβατικών γλωσσών προγραμματισμού πρέπει να έχει και κάποιες δεξιότητες σε συμβολικό προγραμματισμό σε γλώσσες TN, όπως LISP και Prolog, καθώς επίσης και κάποια εμπειρία από τη χρήση διαφορετικών τύπων κελυφών ανάπτυξης ΕΣ.

Ο **διαχειριστής έργου** (project manager) είναι ο ηγέτης της ομάδας ανάπτυξης του ΕΣ, υπεύθυνος για τη διατήρηση της προόδου του έργου. Αυτός/αυτή εξασφαλίζει ότι όλα τα παραδοτέα και τα ορόσημα επιτυγχάνονται, αλληλεπιδρά με τον ειδικό τομέα, το μηχανικό γνώσης, τον προγραμματιστή και τον τελικό χρήστη.

Ο **τελικός χρήστης** (end user) ή απλά χρήστης χρησιμοποιεί το ΕΣ, καθώς αναπτύσσεται. Ο χρήστης πρέπει και να είναι βέβαιος για την απόδοση του συστήματος και να αισθάνεται άνετα με τη χρήση του. Ως εκ τούτου, και ο σχεδιασμός της διεπαφής χρήστη-ΕΣ είναι ζωτικής σημασίας για την επιτυχία του έργου. Η συνεισφορά του τελικού χρήστη εδώ μπορεί να είναι καθοριστική.

5.5 Απόκτηση Γνώσης

Η **Απόκτηση Γνώσης** (Knowledge acquisition) είναι η μεταφορά και η μετατροπή μίας εμπειρίας που αφορά τον τρόπο επίλυσης ενός συγκεκριμένου προβλήματος από την πηγή της γνώσης σε ένα πρόγραμμα (Buchanan et al, 1983). περιλαμβάνει, δε, τρεις δραστηριότητες:

7. Συγκέντρωση γνώσης,
8. Αναπαράσταση γνώσης,
9. Κωδικοποίηση γνώσης.

Η αποκτώμενη γνώση μπορεί να περιέχει αβεβαιότητα ή/και ασάφεια, η διαχείριση των οποίων γίνεται με ειδικούς τρόπους.

5.5.1 Συγκέντρωση Γνώσης

Η **Συγκέντρωση Γνώσης** (Knowledge elicitation) αφορά τη διαδικασία της πρόσκτησης γνώσης από τους ειδικούς τομέα, τα βιβλία, συγγράμματα ή άλλο έντυπο υλικό, από άμεση παρατήρηση των διαδικασιών επίλυσης του προβλήματος από κάποιον ειδήμονα και είναι ένα από τα βασικότερα βήματα για τη δημιουργία ενός ΕΣ. Η συγκέντρωση της γνώσης μπορεί να γίνει και από το ίδιο το σύστημα, το οποίο πολλές φορές έχει τη δυνατότητα παραγωγής νέας γνώσης, οπότε καλείται **ευφυές έμπειρο σύστημα** (intelligent expert system).

Η γνώση που συγκεντρώνεται συγκροτεί τη Βάση Γνώσης, ένα από τα δυο συστατικά τμήματα των ΕΣ. Το δεύτερο συστατικό τμήμα είναι η **συμπερασματική μηχανή**, για τον τρόπο λειτουργίας της οποίας έχουμε διεξοδικά αναφερθεί στο κεφάλαιο 3, καθώς και σε προηγούμενη παράγραφο του κεφαλαίου αυτού.

Η διαδικασία ανάπτυξης της Βάσης Γνώσης ενός ΕΣ καλείται **μηχανική της γνώσης** (knowledge engineering) και οι αρμόδιοι για αυτήν μηχανικοί γνώσης είναι, επίσης, υπεύθυνοι για τη συγκέντρωση της γνώσης από τις διάφορες πηγές, καθώς και την αναπαράστασή της σε μορφή γεγονότων (facts) και κανόνων (rules).

Γνώση = Γεγονότα + Κανόνες
Knowledge = Facts + Rules

Όταν η πηγή συγκέντρωσης είναι ο ειδικός πεδίου, τότε απαιτούνται πολλές προσωπικές συναντήσεις με αυτόν. Στις συναντήσεις αυτές συνήθως απαιτούνται μέθοδοι δομημένης συνέντευξης, η οποία εν γένει αποτυπώνεται σε μαγνητικά μέσα, ώστε η λεπτομερής επεξεργασία της να γίνει αργότερα σε ουδέτερο χρόνο. Με τις δομημένες συνεντεύξεις αποτυπώνεται ειδικού τύπου πληροφόρηση για συγκεκριμένες μορφές

τεχνικών του ειδήμονα που οι απλή προφορική συνέντευξη δε μπορεί να αποτυπώσει. Μέθοδοι μη δομημένων συνεντεύξεων συνήθως ακολουθούνται στις πρώτες συναντήσεις, κατά το προκαταρκτικό στάδιο επαφών, με σκοπό την πιο εύκολη και γρήγορη συγκέντρωση υλικού σχετικού με βασική πληροφόρηση

Για την εύκολη και άμεση άντληση της γνώσης από έναν ειδικό τομέα έχουν αναπτυχθεί πολλά κατάλληλα εξειδικευμένα εργαλεία. Ένα από τα βασικά τους προτερήματα είναι η ευκολία που διαθέτουν, ώστε ο ίδιος ο ειδικός τομέα να συμβάλλει στη ενημέρωση της βάσης γνώσης με την άνεσή του. Τέτοια εργαλεία οδηγούν τον ειδήμονα βήμα προς βήμα για την κατηγοριοποίηση και αποσαφήνιση των γνώσεων τις οποίες πρέπει να μεταδώσει με τέτοιο τρόπο, ώστε να τον βοηθάει να συλλάβει τα στοιχεία της ίδιας του της εμπειρίας. Αφού καταγραφεί η γνώση, οι κανόνες παράγονται αυτόματα.

Άλλο είδος εργαλείων απόκτησης γνώσης είναι τα εργαλεία που παράγουν γνώση επαγωγικά από παραδείγματα (βλέπε κεφάλαιο 4, [αλγόριθμοι παραγωγής κανόνων](#)). Τέτοιο λογισμικό παράγει κανόνες βασιζόμενο σε παραδείγματα τα οποία περιέχουν χαρακτηριστικές λύσεις και αντιπροσωπεύουν την υπό αναζήτηση γνώση.

Όσον αφορά την στρατηγική που ακολουθείται για την απόκτηση γνώσης, αυτή είναι η δημιουργία πρωτοτύπου. Παράγεται, δηλαδή, το γρηγορότερο δυνατό ένα επιδείξιμο σύστημα, για να διατηρηθεί ζωντανό το ενδιαφέρον του πελάτη που το έχει παραγγείλει, ο οποίος βλέποντας γρήγορα ένα αντιπροσωπευτικό σύστημα πιστεύει στη δυνατότητα επίτευξης ενός αξιόλογου τελικού προϊόντος και συνεχίζει να υποστηρίζει την ολοκλήρωση του έργου. Στη πράξη, οι πιο πολλοί μηχανικοί γνώσης χρησιμοποιούν τα γρήγορα πρωτότυπα σε συνδυασμό με συνεντεύξεις και ανάλυση εγγράφων. Εντούτοις, παρουσιάζονται πολλά προβλήματα κυρίως κατά τη φάση της συγκέντρωσης της γνώσης, όπως η χαμηλή ποιότητα της υπό συγκέντρωση γνώσης και προβλήματα σχετικά με τους ειδικούς τομέα: πιο συγκεκριμένα, έλλειψη τρόπων προσέγγισής τους ή επικοινωνίας για μετάδοση της εμπειρίας τους, μη συνεργασία τους.

Όσον αφορά δε το ΕΣ, μπορούμε να πούμε ότι μετά την απόκτηση της γνώσης είναι σχεδόν ολοκληρωμένο, με μόνη εκκρεμότητα τη δημιουργία της διεπαφής που μπορεί να προγραμματιστεί και με άλλα εξειδικευμένα εργαλεία, π.χ. γλώσσες προγραμματισμού κατάλληλες για διαδικτυακές εφαρμογές ή κινητές συσκευές, όταν το ΕΣ λειτουργεί σε τέτοια περιβάλλοντα.

5.6 Αβεβαιότητα

Αβεβαιότητα (Uncertainty) είναι η έλλειψη επαρκούς πληροφόρησης για τη λήψη μίας απόφασης. Μπορεί να οφείλεται σε αβέβαιη γνώση ή αβέβαιη μαρτυρία, η οποία οδηγεί σε αλληλοσυγκρουόμενες λύσεις ή μη ολοκληρωμένη γνώση. Τα ΕΣ πρέπει να διαθέτουν ικανότητες παρόμοιες με εκείνες των ανθρώπων, που είναι ικανοί να εξάγουν αληθή συμπεράσματα αντιμετωπίζοντας αβέβαιη πληροφόρηση. Η ικανότητα αυτή των ειδημόνων βασίζεται στη χρήση απαγωγικής συλλογιστικής σε μορφή ευρετικών κανόνων (heuristics): κανόνων που σχηματίζει ο ειδικός τομέα από συνεχή παρατήρηση παρόμοιων παρελθοντικών καταστάσεων και λύσεων που βρέθηκαν. Παράδειγμα, όταν κάποιος οδηγός βρεθεί σε σταυροδρόμι και δεν ξέρει ποιο δρόμο να ακολουθήσει, συνήθως επιλέγει αυτόν που έχει κατεύθυνση προς το στόχο του, γιατί τις πιο πολλές φορές αυτό έχει αποδειχθεί σωστή επιλογή. Πολλές φορές, οι ειδήμονες χρησιμοποιούν τον ευρετικό τρόπο αιτιολόγησης και ως μία τεχνική επίλυσης προβλημάτων, όταν θέλουν γρήγορα και αποτελεσματικά να φτάσουν σε μία καλή λύση. Όταν χρησιμοποιούνται ευρετικές τεχνικές, η διαδικαστική γνώση περιγράφεται με κανόνες περιορισμένης αποδοχής με τους οποίους μπορεί να δοθεί γρήγορα μία «πιθανή», καλή και αιτιολογημένη λύση. Οι ευρετικές μέθοδοι μπορούν να προσφέρουν μεθόδους που μειώνουν αισθητά το χρόνο επεξεργασίας και το κόστος επίλυσης ενός προβλήματος, αλλά δεν εγγυώνται την εύρεση λύσης, όπως την εγγυάται μια μέθοδος βασισμένη σε βέβαιη γνώση.

Τα ΕΣ είναι ικανά να χειρίζονται τρεις τύπους μη ακριβούς πληροφορίας:

- Αβεβαιότητα που προκύπτει, όταν κάποιος δεν είναι απόλυτα βέβαιος για κάποια πληροφορία· παράδειγμα:

Ίσως ο Γιάννης είναι άρρωστος (είναι ή όχι);

- Ασάφεια που προκύπτει, όταν κάποια πληροφορία δεν είναι σαφώς οριοθετημένη· παράδειγμα:

Ο Γιάννης είναι ψηλός (δηλαδή πόσο;)

- Συνδυασμός των δύο προηγούμενων τύπων· παράδειγμα:

Ο Γιάννης είναι μάλλον ψηλός (είναι ψηλός και πόσο;)

Τα ΕΣ χρησιμοποιούν διάφορα μοντέλα για τη διαχείριση της μη ακριβούς πληροφορίας, εκ των οποίων γνωστότερα είναι τα ακόλουθα δυο τα οποία αναλύουμε στα επόμενα κεφάλαια:

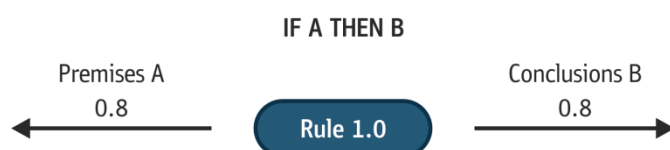
- Συντελεστές βεβαιότητας,
- Ασαφής λογική.

5.6.1 Συντελεστές Βεβαιότητας

Όταν υπάρχει αβεβαιότητα ως προς την αλήθεια μιας πληροφορίας, χρειάζεται ένας τρόπος για να εκφραστεί το ποσοστό πίστης σε αυτήν. Ένας κατάλληλος τρόπος είναι να υιοθετήσουμε το μοντέλο των συντελεστών βεβαιότητας, το οποίο σε ένα ΕΣ εξετάζει την αβεβαιότητα και στους ισχυρισμούς των κανόνων και την εγκυρότητα των συμπερασμάτων που εξάγονται κατά την εκτέλεσή του.

Πιο αναλυτικά, ένας κανόνας του τύπου [IF <premises> THEN <conclusions>] μπορεί να περιέχει αβεβαιότητα στην ακρίβεια της αλήθειας των γεγονότων που αποτελούν τους ισχυρισμούς του και στην εγκυρότητα των συμπερασμάτων στα οποία καταλήγει. Επιπλέον, όταν ένα ΕΣ περιέχει αβέβαιη γνώση, τότε αβεβαιότητα θα περιέχει και το τελικό συμπέρασμα το οποίο θα εξαγάγει το ΕΣ μετά την αλυσιδωτή εκτέλεση των αβέβαιων κανόνων του.

Στο σχήμα 5.5 παρουσιάζεται ένας κανόνας 100% βέβαιος ως προς την ορθότητα των συμπερασμάτων του (συντελεστής βεβαιότητας 1.0) αλλά 80% βέβαιος ως προς την ακρίβεια της αλήθειας των ισχυρισμών του (συντελεστής βεβαιότητας 0.8). Στην περίπτωση αυτή, δεχόμαστε την εγκυρότητα του κανόνα με την ίδια βεβαιότητα που περιέχεται στους ισχυρισμούς, δηλαδή 80%. Για τον παραπάνω υπολογισμό χρησιμοποιήθηκε το μοντέλο υπολογισμού των συντελεστών βεβαιότητας.



Σχήμα 5.5 Κανόνας με βεβαιότητα 1.0

Στη συνέχεια παρουσιάζεται ολοκληρωμένα ο τρόπος υπολογισμού του βαθμού βεβαιότητας για το αποτέλεσμα ενός ΕΣ που περιέχει αβέβαιη γνώση με βάση το μοντέλο των συντελεστών βεβαιότητας.

Οι **συντελεστές βεβαιότητας** (certainty factors-CF) χρησιμοποιούνται, για να αναπαραστήσουν βάρη που έχουν καθοριστεί με ευρετικό τρόπο και τα οποία προσδιορίζουν το καθαρό μέτρο εμπιστοσύνης σε μία αξία, γεγονός ή δεδομένο με τιμές μεταξύ -1 και +1. Σε ένα από τα γνωστότερα πρώτο-δημιουργημένα ΕΣ, το MYCIN, ο συντελεστής βεβαιότητας ορίζεται ως εξής:

$$CF = \frac{MB - MD}{1 - \min(MB, MD)}$$

όπου MB = το **μέτρο εμπιστοσύνης** (measure of belief)
και MD = το **μέτρο της μη εμπιστοσύνης** (measure of disbelief).

Τα MB και MD ορίζονται και τα δύο από τους ειδήμονες.

Σε τέτοιου είδους προσεγγίσεις, οι ανεξάρτητοι συντελεστές βεβαιότητας των διαφόρων ισχυρισμών και κανόνων συνδυάζονται, ώστε να καταλήξουν σε κοινό συντελεστή για το συμπέρασμα το οποίο συνάγει το σύστημα. Αυτό είναι ανεπίσημος τρόπος καταμέτρησης της εμπιστοσύνης του ειδικού τομέα και ταιριάζει απόλυτα σε εφαρμογές όπου υπάρχει αβεβαιότητα, αλλά όχι ασάφεια.

Γενικά, κάθε πρόταση ενός μέρους AN ενός κανόνα μπορεί να έχει το δικό της συντελεστή βεβαιότητας, όπως μπορεί να έχει και κάθε πρόταση του μέρους TOTE του κανόνα. Ο συντελεστής βεβαιότητας της σύζευξης δυο ή περισσότερων ισχυρισμών (A και B και ... και X) ορίζεται ως ο μικρότερος των συντελεστών των μεμονωμένων ισχυρισμών.

$$CF_{if-part(AND)} = \min CF_{premisses}$$

Ο συντελεστής βεβαιότητας μιας διάζευξης δυο ή περισσότερων ισχυρισμών (A ή B ή ... ή Y) ορίζεται ως ο μεγαλύτερος των συντελεστών των μεμονωμένων ισχυρισμών.

$$CF_{if-part(OR)} = \max CF_{premisses}$$

Ο συντελεστής βεβαιότητας ενός κανόνα ορίζεται ως ο συντελεστής του μέρους AN πολλαπλασιασμένος με τον συντελεστή του μέρους TOTE .

$$CF_{rule} = CF_{if-part} * CF_{then-part}$$

Παράδειγμα εφαρμογής συντελεστών βεβαιότητας:

AN τα φρένα είναι σπασμένα (CF=0.4)
και το τιμόνι δε λειτουργεί (CF=0.2)
TOTE θα γίνει δυστύχημα (CF=0.9)

$$\begin{aligned} CF_{rule} &= CF_{if-part} * CF_{then-part} = \\ &= \min CF_{premisses} * CF_{then-part} = \\ &= 0.2 * 0.9 = 0.18 \end{aligned}$$

Δεν επιτρέπεται να ενεργοποιούνται κανόνες με συνολικό συντελεστή βεβαιότητας μικρότερο από μια συγκεκριμένη τιμή, π.χ. 0.2. Έτσι, μειώνεται η πιθανότητα πυροδότησης ακατάλληλων κανόνων που θα οδηγούσε σε μείωση της αποτελεσματικότητας του συστήματος. Στο παράδειγμά μας ο συγκεκριμένος κανόνας δεν πρόκειται να εκτελεστεί λόγω του μικρού συντελεστή βεβαιότητάς του.

Ο συντελεστής βεβαιότητας ενός γεγονότος που παράγεται ως συμπέρασμα ενός κανόνα ή μιας αλυσίδας κανόνων, είναι ο μικρότερος των επιμέρους συντελεστών βεβαιότητας των κανόνων που αποτελούν την αλυσίδα:

$$CF_{fact} = \min(CF_{rule_1}, \dots, CF_{rule_n})$$

Παράδειγμα υπολογισμού συντελεστή βεβαιότητας τελικού συμπεράσματος ΕΣ:

Κανόνας1: Εάν η κασέτα δεν κινείται
και η συσκευή δεν ηχογραφεί
και δεν υπάρχει φωτεινό σήμα
Τότε δεν έχει ενεργοποιηθεί η συσκευή (πιθανότητα 0.85)

Κανόνας2: Εάν η ταχύτητα είναι ασταθής
και ο ήχος είναι διαταραγμένος
και υπάρχουν παράσιτα
Τότε η κεφαλή είναι βρώμικη (πιθανότητα 0.75)

Κανόνας3: Εάν δεν υπάρχει φωτεινό σήμα
και η κασέτα δεν κινείται
και η συσκευή δεν ηχογραφεί
Τότε η συσκευή είναι στην παύση (πιθανότητα 0.75)

Κανόνας4: Εάν η κεφαλή είναι βρώμικη
Τότε καθάρισε την κεφαλή και θα αντιμετωπιστεί το πρόβλημα
(πιθανότητα 0.8)

Γεγονότα

η ταχύτητα είναι ασταθής (0.8)
ο ήχος είναι διαταραγμένος (0.6)
υπάρχουν παράσιτα (0.9)
υπάρχει φωτεινό σήμα (1.0)

Οι κανόνες που ενεργοποιούνται κατά σειρά είναι :

1) ο κανόνας 2 με συντελεστή βεβαιότητας συμπεράσματος (η κεφαλή είναι βρώμικη):

$$CF_{\text{κανόνας2}} = \min CF_{\text{premises}} \cdot CF_{\text{then-part}} = \min(0.8 * 0.6 * 0.9) * 0.75 = 0.6 * 0.75 = 0.45$$

2) ο κανόνας 4 με συντελεστή βεβαιότητας συμπεράσματος (καθάρισε την κεφαλή και θα αντιμετωπιστεί το πρόβλημα):

$$CF_{\text{κανόνας4}} = CF_{\text{premises}} * CF_{\text{then-part}} = 0.45 * 0.8 = 0.36$$

Άρα η βεβαιότητα με την οποία προτείνεται να εφαρμοστεί το τελικό συμπέρασμα του ΕΣ είναι 0.36.

5.7 Ασάφεια

Η αρχή της **ασάφειας** (fuzziness) δηλώνει ότι τα πάντα είναι ζήτημα βαθμού. Μόνο από τον κόσμο των μαθηματικών έχει εξαλειφθεί η ασάφεια. Το επίσημο επιστημονικό όνομα της ασάφειας είναι **πολυτιμία** ή **πλειοτιμία**. Η **δίτιμία** (το άσπρο-μαύρο, ναι-όχι, αληθές-ψευδές) απλουστεύουν τα μαθηματικά και τις διαδικασίες των ηλεκτρονικών υπολογιστών. Όμως, η δίτιμη λογική οδηγεί σε πολλά παράδοξα που μόνο η πολυτιμία μπορεί να αντιμετωπίσει.

Μια διαφορετική προσέγγιση στη διαχείριση της αβεβαιότητας και ειδικότερα της ασαφούς μορφής της είναι η έννοια των ασαφών συνόλων, που πρωτοαναφέρθηκαν από τον Zadeh (1965). Στα παραδοσιακά συστήματα έχουμε σαφώς καθορισμένα περιθώρια· για παράδειγμα, τα όρια των ακέραιων αριθμών μεταξύ 3 και 6. Αντίθετα, τα **ασαφή σύνολα** (fuzzy sets) είναι μια κλάση αντικειμένων με όχι απόλυτα καθορισμένα όρια, π.χ.

Οι Έλληνες είναι κοντοί

Το ερώτημα που τίθεται είναι: ο χαρακτηρισμός ‘κοντός’ τι όρια έχει, όταν αναφέρεται σε Έλληνες και, ακόμα περισσότερο, πώς θα διαμορφώνονταν αυτά τα όρια, αν αναφέρονταν σε κάποιον άλλον λαό, όπως οι Κινέζοι ή οι Νορβηγοί; Παρ’ όλη, όμως, την ελαστικότητα αυτών των ορίων, το νόημα της πρότασης γίνεται σαφώς αντιληπτό.

Η γνώση της ασάφειας συνοψίζεται σε **ασαφείς κανόνες** (fuzzy rules). Ένας ασαφής κανόνας συσχετίζει ασαφείς έννοιες σε μια μορφή υποθετικών προτάσεων, π.χ.

Εάν η κίνηση είναι **πυκνή**,
 διατήρησε το πράσινο φως **περισσότερο** χρόνο

Οι προτάσεις *κίνηση είναι πυκνή* και *περισσότερος χρόνος υποδηλώνουν* η καθεμία ένα ασαφές σύνολο.

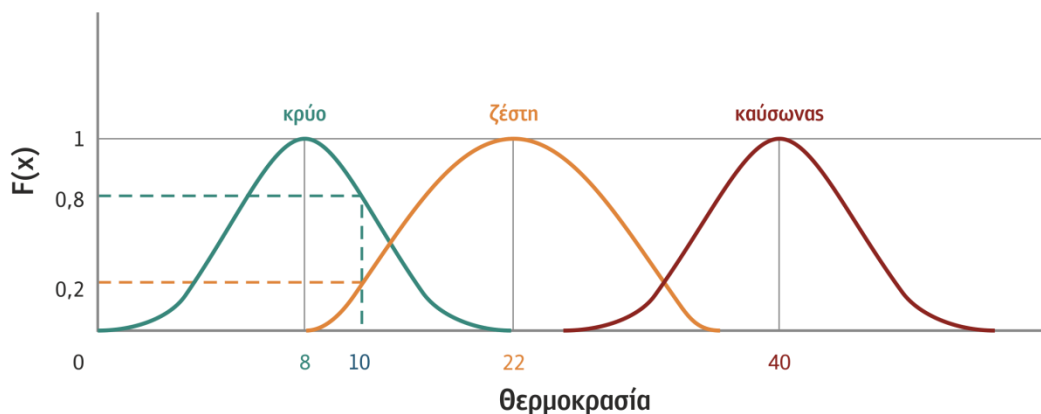
Στην **ασαφή λογική** (fuzzy logic), τα στοιχεία κάθε συνόλου είναι προσαρτημένα στο σύνολο με βαθμούς συμμετοχής, οι οποίοι προκύπτουν μέσω μιας συνάρτησης συμμετοχής και υποδεικνύουν σε ποια έκταση το κάθε στοιχείο είναι μέρος του συνόλου.

Συνάρτηση συμμετοχής (membership function) καλείται η διασπορά των βαθμών συμμετοχής σε ένα ασαφές σύνολο. Η συνάρτηση συμμετοχής F είναι μια συνάρτηση από ένα χώρο στο διάστημα $[0,1]$, όπου

- $F(x)=0$ σημαίνει ότι το x δεν είναι στοιχείο του συνόλου,
- $F(x)=1$ σημαίνει ότι το x είναι οριστικά μέρος του συνόλου,
- όλες οι υπόλοιπες ενδιάμεσες τιμές δείχνουν τους διάφορους βαθμούς συμμετοχής του x στο σύνολο.

Τις περισσότερες φορές οι βαθμοί συμμετοχής καθορίζονται υποκειμενικά από κάποιον ειδικό τομέα.

Ένα από τα στοιχεία της ασαφούς λογικής είναι η έννοια των γλωσσολογικών μεταβλητών, οι οποίες είναι άμεσα εξαρτώμενες από την έννοια που εκπροσωπούν· παράδειγμα, η μεταβλητή *θερμοκρασία* θα μπορούσε να δέχεται ως *γλωσσολογικές μεταβλητές* τα «κρύο», «ζέστη», «καύσωνας», κλπ. Κάθε γλωσσολογική μεταβλητή μπορεί να είναι ασαφής και να έχει δικούς της βαθμούς συμμετοχής στο σύνολο, όπως φαίνεται στο επόμενο σχήμα:



Σχήμα 5.6 Παράδειγμα βαθμών συμμετοχής γλωσσικών μεταβλητών

Όπου: $F_{\text{κρύο}}(8)=1.0$, $F_{\text{κρύο}}(10)=0.8$, $F_{\text{κρύο}}(40)=0.0$
 $F_{\text{ζέστη}}(10)=0.2$, $F_{\text{ζέστη}}(22)=1.0$, $F_{\text{καύσωνας}}(40)=1.0$

Δεδομένων δυο συνόλων A και B , η συνάρτηση συμμετοχής μιας ένωσης συνόλων είναι:

$$F_{A \text{ and } B}(x) = \min(F_A(x), F_B(x))$$

Για παράδειγμα, η τιμή 10 που παρατηρήθηκε ως θερμοκρασία συμμετέχει στην ένωση των ασαφών γλωσσολογικών μεταβλητών 'ζέστη' και 'κρύο' ως εξής:

$$F_{\text{ζέστη και κρύο}}(10) = \min(F_{\text{ζέστη}}(10), F_{\text{κρύο}}(10)) = \min(0.8, 0.2) = 0.2$$

Η συνάρτηση συμμετοχής μιας ένωσης ασαφών εννοιών υπολογίζεται ως ακολούθως:

$$F_{A \text{ or } B}(x) = \max(F_A(x), F_B(x))$$

Για παράδειγμα:

$$F_{\text{ζέστη ή καύσωνας}}(22) = \max(F_{\text{ζέστη}}(22), F_{\text{καύσωνας}}(22)) = \max(1.0, 0.2) = 1.0$$

Ως συνάρτηση συμμετοχής συνήθως χρησιμοποιείται μια σιγμοειδής (sigmoid), τριγωνική (triangular) ή τραπεζοειδής ή (trapezoid) συνάρτηση.

5.7.1 Ασαφείς κανόνες

Όπως προαναφέραμε, ένας ασαφής κανόνας) συσχετίζει ασαφείς έννοιες σε μια μορφή υποθετικών προτάσεων:

ΕΑΝ ο καιρός είναι **κρύος** και ντυθείς **ελαφρά**
ΤΟΤΕ θα αρρωστήσεις **πολύ**

ΑΝ αρρωστήσεις **πολύ**
ΤΟΤΕ δε θα γράψεις **καλά** στις εξετάσεις

Η αναλυτική περιγραφή ενός ασαφούς κανόνα είναι μία ασαφής σχέση $R(x,y)$ που ονομάζεται **σχέση συνεπαγωγής** (implication relation).

Υποθέστε τώρα ότι έχουμε τους ακόλουθους κανόνες:

Κανόνας1: Εάν E_{11} και E_{12}
Τότε H_1

Κανόνας2: Εάν E_{21} και E_{22}
Τότε H_2

Κανόναςn:....

Για τον προσδιορισμό της *τιμής αληθείας* (truth value) που αφορά όλους τους κανόνες που εκτελούνται σε ένα σύστημα. μπορούμε να χρησιμοποιήσουμε την **max-min** μέθοδο, όπου η τελική τιμή αληθείας δίνεται από τη συνάρτηση:

$$F_{\text{αληθείας}} = \max \left[\left(\min(F_{11}, F_{12},) \right), \min(F_{21}, F_{22},), \dots \right]$$

Εκτός της max-min συνάρτησης, υπάρχουν και άλλες λιγότερο ή περισσότερο πολύπλοκες.

Παράδειγμα εφαρμογής Ασαφούς λογικής

Χρησιμοποιήστε τη μέθοδο max-min που παρουσιάστηκε ακριβώς από πάνω, για να βρείτε την τιμή αληθείας στο επόμενο σύστημα κανόνων, που αφορά την κίνηση μιας βαλβίδας σε σχέση με τη θερμοκρασία και την πίεση σε ένα μηχάνημα:

Κανόνας1: Εάν η θερμοκρασία είναι κρύα
και η πίεση είναι αδύνατη
Τότε η κίνηση της βαλβίδας είναι θετικά υψηλή (PL)

Κανόνας2: Εάν η θερμοκρασία είναι κρύα και η πίεση είναι χαμηλή Τότε η κίνηση της βαλβίδας είναι θετικά μέτρια (PM)

Κανόνας3: Εάν η θερμοκρασία είναι κρύα και η πίεση είναι OK Τότε η κίνηση της βαλβίδας είναι μηδενική (ZR)

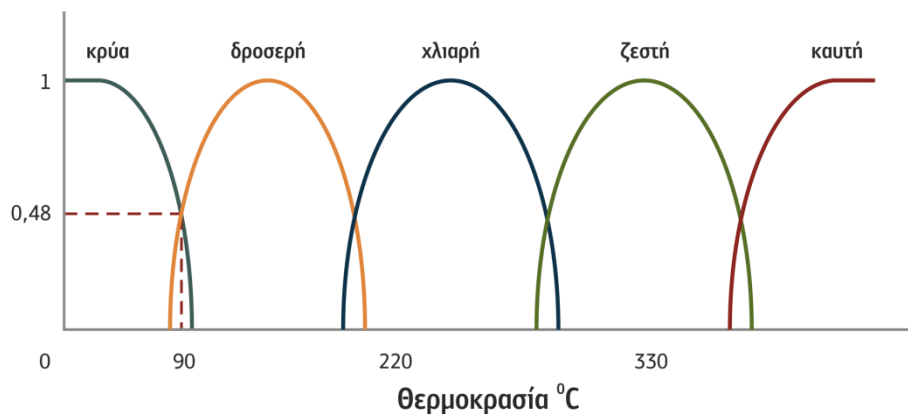
Κανόνας4: Εάν η θερμοκρασία είναι κρύα και η πίεση είναι δυνατή Τότε η κίνηση της βαλβίδας είναι αρνητικά μέτρια (NM)

Η παρατήρηση των συνθηκών που επικρατούν στο μηχάνημα είναι οι ακόλουθες:

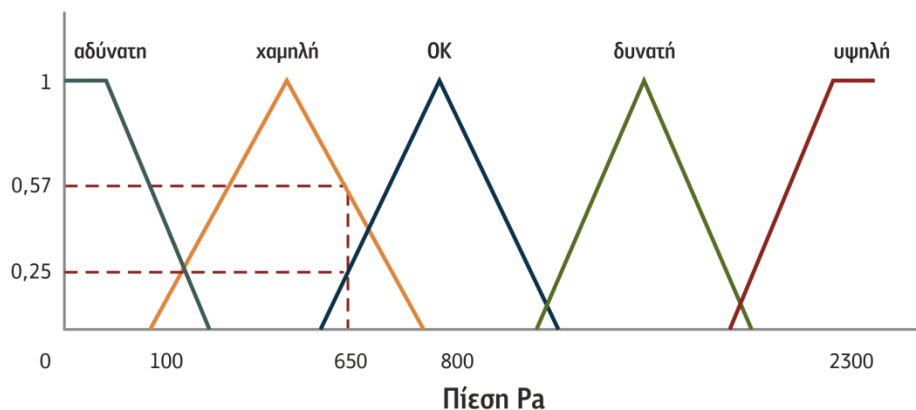
Θερμοκρασία: 90°

Πίεση: 650Pa

Ακολουθεί η διαγραμματική παρουσίαση των συναρτήσεων συμμετοχής των γλωσσολογικών μεταβλητών της θερμοκρασίας και της πίεσης, όπου με κόκκινη διακεκομμένη μεταβλητή σημειώνονται οι βαθμοί συμμετοχής της παρατηρούμενης θερμοκρασίας και πίεσης.



Σχήμα 5.7 Βαθμός συμμετοχής της τιμής 90° για τη θερμοκρασία



Σχήμα 5.8 Βαθμός συμμετοχής της τιμής 650Pa για την πίεση

Βάσει των δεδομένων, οι κανόνες που εκτελούνται είναι ο 2 και ο 3.

Κανόνας 2 :

$$\begin{aligned}
 F_{\text{θερμοκρασία κρύα and πίεση χαμηλή}}(90^0, 650Pa) \\
 &= \min(F_{\text{θερμοκρασία κρύα}}(90^0), F_{\text{πίεση χαμηλή}}(650Pa)) \\
 &= \min(0.48, 0.57) = 0.48
 \end{aligned}$$

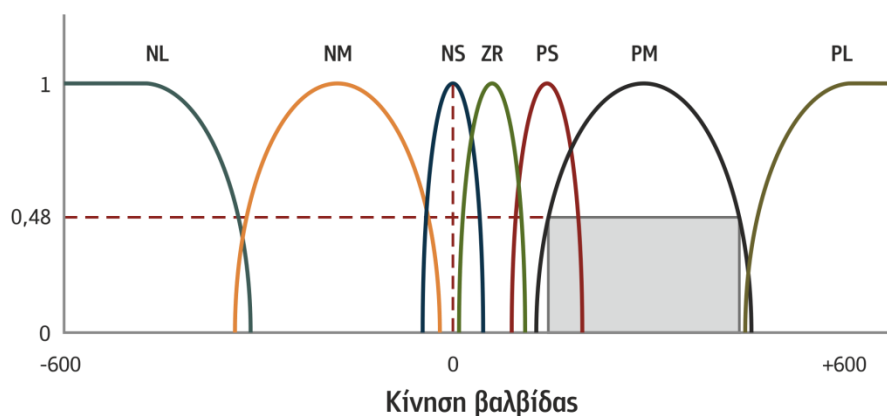
Κανόνας 3 :

$$\begin{aligned}
 F_{\text{θερμοκρασία κρύα and πίεση OK}}(90^0, 650Pa) \\
 &= \min(F_{\text{θερμοκρασία κρύα}}(90^0), F_{\text{πίεση OK}}(650Pa)) \\
 &= \min(0.48, 0.25) = 0.25
 \end{aligned}$$

Επομένως, η τιμή αληθείας και για τους δυο κανόνες υπολογίζεται με τη μέθοδο max -min ως εξής:

$$\begin{aligned}
 F_{\text{αληθείας}} \\
 &= \max[\min(F_{\text{θερμοκρασία κρύα}}(90^0), F_{\text{πίεση χαμηλή}}(650Pa)), \\
 &\quad (F_{\text{θερμοκρασία κρύα}}(90^0), F_{\text{πίεση OK}}(650Pa))] \\
 &= \max[0.48, 0.25] = 0.48
 \end{aligned}$$

Η συνάθροιση των επιμέρους αποτελεσμάτων οδηγεί στην τιμή αληθείας 0.48 για την κίνηση της βαλβίδας που αποτυπώνεται γραφικά στο επόμενο σχήμα.



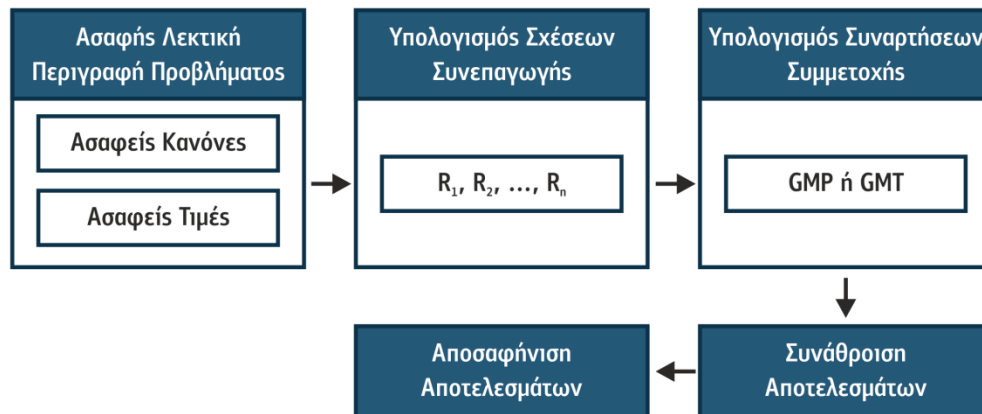
Σχήμα 5.9 Βαθμός συμμετοχής την κίνηση της βαλβίδας

Επομένως, η προτεινόμενη τελικώς κίνηση της βαλβίδας είναι αυτή που υποδεικνύει ο κανόνας 2 που παρέχει τη μεγαλύτερη τιμή αλήθειας, δηλαδή ότι η κίνηση της βαλβίδας πρέπει να είναι θετικά μέτρια (PM).

Οι παραπάνω υπολογισμοί παρουσιάστηκαν αρκετά στοιχειωδώς. Η εξαγωγή συμπερασμάτων με ασαφή λογική είναι αρκετά πιο πολύπλοκη, γίνεται σε 4 στάδια (βλέπε Σχήμα 5.10) και ολοκληρώνεται με την αποσαφήνιση των αποτελεσμάτων που προκύπτουν κατά τα 3 πρώτα στάδια:

- **Στάδιο 1:** Υπολογισμός του βαθμού συμμετοχής (σχέση συνεπαγωγής) στο ασαφές σύνολο κάθε εμπλεκόμενου ασαφούς δεδομένου .

- **Στάδιο 2:** Παραγωγή επιμέρους αποτελεσμάτων κανόνων μέσω κάποιας συλλογιστικής διαδικασίας.
- **Στάδιο 3:** Συνάθροιση των επιμέρους αποτελεσμάτων.
- **Στάδιο 4:** Αποσαφήνιση αποτελεσμάτων.



Σχήμα 5.10 Τα 4 στάδια εξαγωγής συμπερασμάτων με ασαφή λογική

5.8 Προτερήματα και Μειονεκτήματα Εμπείρων Συστημάτων Βασισμένων σε Κανόνες

Ως πλεονεκτήματα των ΕΣ έναντι άλλων τύπων συστημάτων επίλυσης προβλημάτων μπορούν να θεωρηθούν τα παρακάτω:

- Παρέχουν φυσική αναπαράσταση γνώσης. Ένας ειδικός πεδίου εξηγεί συνήθως τη διαδικασία επίλυσης ενός προβλήματος με εκφράσεις όπως αυτή: «Κάτω από τέτοιες-και-τέτοιες συνθήκες, το κάνω-και-έτσι". Τέτοιου τύπου εκφράσεις μπορούν εύκολα να διατυπωθούν αρκετά φυσικά με τη βοήθεια κανόνων παραγωγής.
- Διαθέτουν ομοιόμορφη δομή. Όλοι οι κανόνες παραγωγής έχουν την ίδια δομή AN –TOTE. Κάθε κανόνας είναι ανεξάρτητο κομμάτι της γνώσης. Η ίδια η σύνταξη των κανόνων παραγωγής τούς επιτρέπει να είναι αυτο-τεκμηριωμένοι.
- **Διαχωρίζουν την αναπαράσταση της γνώσης από την επεξεργασία της.** Η δομή ενός ΕΣ Βασισμένου σε Κανόνες παρέχει αποτελεσματικό διαχωρισμό της βάσης των γνώσεων από τη Συμπερασματική Μηχανή. Έτσι, καθίσταται δυνατή η ανάπτυξη διαφορετικών εφαρμογών με χρήση του ίδιου κελύφους ΕΣ.
- **Είναι δυνατή η ενασχόληση με ελλιπή και αβέβαιη γνώση.** Τα περισσότερα ΕΣ Βασισμένα σε Κανόνες είναι σε θέση να διαχειρίζονται και ατελή και αβέβαιη γνώση.

Μειονεκτήματα μπορεί να θεωρηθούν τα ακόλουθα:

- **Οι σχέσεις μεταξύ κανόνων είναι αδιαφανείς.** Παρά το γεγονός ότι οι επιμέρους κανόνες παραγωγής είναι σχετικά απλοί και αυτο-τεκμηριωμένοι, οι λογικές αλληλεπιδράσεις μεταξύ τους, στο μεγάλο σύνολο των κανόνων, μπορεί να είναι αδιαφανείς. Στα ΕΣ Βασισμένα σε Κανόνες είναι δύσκολο να παρατηρήσουμε πώς οι μεμονωμένοι κανόνες εξυπηρετούν τη συνολική στρατηγική.
- **Η στρατηγική αναζήτησης είναι αναποτελεσματική.** Η Συμπερασματική Μηχανή εφαρμόζει εξαντλητική αναζήτηση μέσω όλων των κανόνων παραγωγής κατά τη διάρκεια

κάθε κύκλου. ΕΣ με ένα μεγάλο σύνολο κανόνων (>100 κανόνες) χρειάζονται μεγάλο χρόνο εκτέλεσης και, συνεπώς, μπορεί να είναι ακατάλληλα για εφαρμογές πραγματικού χρόνου.

- Ένα τυπικό ΕΣ έχει αδυναμία να μάθει. Σε γενικές γραμμές, τα ΕΣ Βασισμένα σε Κανόνες δεν έχουν την ικανότητα να μαθαίνουν από την εμπειρία. Σε αντίθεση με έναν εμπειρογνώμονα, ο οποίος ξέρει πότε να «παραβιάσει τους κανόνες», ένα ΕΣ δεν μπορεί να τροποποιήσει αυτόματα τη βάση των γνώσεών του ή να προσαρμόσει τους ισχύοντες κανόνες ή να προσθέσει νέους. Ο μηχανικός γνώσης εξακολουθεί να είναι υπεύθυνος για την αναθεώρηση και διατήρηση του συστήματος.

Βιβλιογραφία/Αναφορές

Buchanan, B.G. (1983). Partial bibliography of work on expert systems. *Sigart Newsletter*, 84, 45-50.

Ανακτήθηκε από <http://infolab.stanford.edu/pub/cstr/reports/cs/tr/82/953/CS-TR-82-953.pdf>

Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems* (2^η έκδοση). Έσσεξ, Αγγλία: Pearson Education.

Russel, S. & Norvig P. (2003). *Τεχνητή Νοημοσύνη - Μια Σύγχρονη Προσέγγιση* (2^η έκδοση). Αθήνα: Εκδόσεις Κλειδάριθμος.

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, 338-353.

Ανακτήθηκε από <http://www.cs.berkeley.edu/~zadeh/papers/Fuzzy%20Sets-Information%20and%20Control-1965.pdf>

Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κοκκοράς, Φ., & Σακελλαρίου Η. (2011). *Τεχνητή Νοημοσύνη* (3^η έκδοση). Εκδόσεις Πανεπιστημίου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Ποια από τα παρακάτω αποτελούν προτερήματα των ΕΣ

1. Αδιαφανείς σχέσεις μεταξύ κανόνων.
2. Φυσική αναπαράσταση γνώσης.
3. Διαχωρισμό της αναπαράστασης της γνώσης από την επεξεργασία της.
4. Μη τροποποιήσιμη βάση κανόνων.
5. Ενασχόληση με ελλιπή και αβέβαιη γνώση.

Απάντηση/Λύση

Σωστή απάντηση είναι τα 2,3 και 5.

Κριτήριο αξιολόγησης 2

Στα έμπειρα συστήματα, η υπάρχουσα «εμπειρία» για το πρόβλημα βρίσκεται αποτυπωμένη:

1. Στα δηλωμένα γεγονότα (facts).

2. Στους κανόνες που βρίσκονται στην ατζέντα σε κάθε στιγμιότυπο.
3. Στους κανόνες που βρίσκονται αποθηκευμένοι στη βάση γνώσης.
4. Σε όλα τα προηγούμενα.

Απάντηση/Λύση

Σωστή απάντηση είναι το 3.

ΚΕΦΑΛΑΙΟ 6 - Νοήμονες Πράκτορες

Σύνοψη

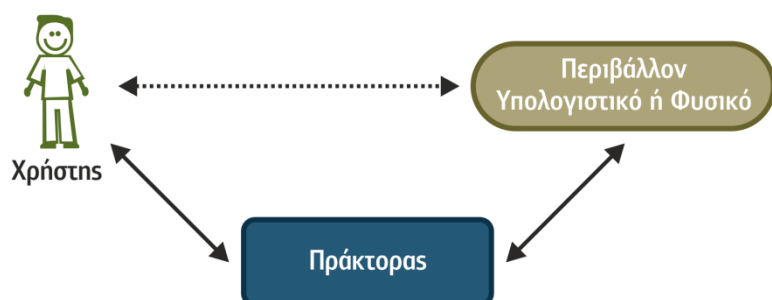
Οι νοήμονες (ή ευφυείς) πράκτορες αποτελούν σύγχρονα συστήματα Τεχνητής Νοημοσύνης στα οποία δυνητικά μπορούν να χρησιμοποιηθούν επιλεκτικά και σε συνδυασμό μέθοδοι αναπαράστασης γνώσης και επίλυσης προβλημάτων με τεχνολογίες που παρουσιάστηκαν στα προηγούμενα κεφάλαια. Με αυτό ως γνώμονα παρουσιάζονται οι διαφορετικές βασικές αφηρημένες αρχιτεκτονικές των νοημών πρακτόρων, γίνεται αναφορά στα πολυπρακτορικά συστήματα και παρουσιάζονται εφαρμογές τους.

Προαπαιτούμενη γνώση

Επίλυση Προβλημάτων, Αναπαράσταση Γνώσης, Έμπειρα Συστήματα, Μηχανική μάθηση.

6.1 Εισαγωγικά

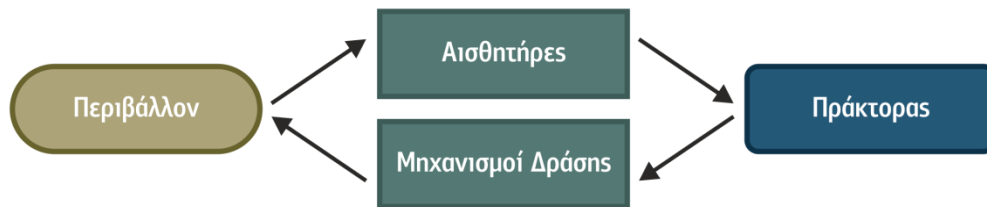
Η εμφάνιση των **πρακτόρων** (agents) και ειδικότερα των **ευφύων πρακτόρων** (intelligent agents) στον κλάδο της ΤΝ και, γενικότερα, στην περιοχή της Επιστήμης των Υπολογιστών υπόσχεται ριζικές αλλαγές στην επικοινωνία μεταξύ χρήστη και λογισμικού στο σημερινό διασυνδεδεμένο και δικτυωμένο ψηφιακό κόσμο. Ήδη, γίνεται αισθητή η παρουσία τους σε πληθώρα εφαρμογών, όπως είναι η αναζήτηση και το φιλτράρισμα των πληροφοριών στο διαδίκτυο, η παροχή έξυπνων υπηρεσιών βοήθειας και εξυπηρέτησης πελατών, καθώς και ο έλεγχος σωστής λειτουργίας μεγάλων εργοστασιακών μονάδων.



Σχήμα 6.1 Διασύνδεση χρήστη-προγράμματος μέσω πράκτορα

Η έννοια του όρου *πράκτορας* παραπέμπει στο πρόσωπο το οποίο ενεργεί εκ μέρους ενός άλλου προσώπου, δηλαδή σε αυτό που καλούμε *αντιπρόσωπο*, εναλλακτικά *πράκτορα* ή *μεσάζοντα*. Γενικότερα, η χρήση του όρου *αντιπρόσωπος* δεν είναι δόκιμη, δεδομένου ότι ένας αντιπρόσωπος απλώς φέρει εις πέρας ένα πλήθος εντολών. Αντίθετα, ένας πράκτορας είναι σε θέση να εκπληρώσει ένα πλήθος στόχων με την ελάχιστη δυνατή εμπλοκή του χρήστη.

Έχει αποδειχθεί ιδιαίτερα δύσκολο το να δοθεί ένας αυστηρός ορισμός για τον όρο *πράκτορας* λόγω του ότι ορισμένα από τα χαρακτηριστικά του έχουν τελείως διαφορετική σημασία και βαρύτητα στους διαφορετικούς επιστημονικούς κλάδους που ερευνούν το αντικείμενο. Σε μία, όσο γίνεται, γενικότερη θεώρηση, ένας πράκτορας είναι μία οντότητα που αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται με τη βοήθεια *αισθητήρων* (*sensors*), αποτελεί μέρος του περιβάλλοντος αυτού, κάνει συλλογισμούς και δρα σε αυτό με τη βοήθεια *μηχανισμών δράσης* (*effectors*) για την επίτευξη δεδομένων στόχων εκ μέρους του χρήστη ή κάποιας άλλης οντότητας λογισμικού που πρέπει να διαθέτει κάποιο βαθμό αυτονομίας (Russel & Norvig, 2009).



Σχήμα 6.2 Βασικός Πράκτορας κατά Russel και Norvig

Αν και δεν είναι εφικτό να δοθεί, όπως προείπαμε, ένας ενιαίος ορισμός για όλα τα είδη πρακτόρων, δεδομένου ότι διαφέρουν μεταξύ τους στον τρόπο λειτουργίας τους και το επιστημονικό πεδίο στο πλαίσιο του οποίου καλούνται να λειτουργήσουν, ο παραπάνω βασικός ορισμός κατά Russel και Norvig αποτελεί κοινό τους παρανομαστή.

Ο βασικός αυτός ορισμός περιγράφει έναν άνθρωπο, ένα ρομπότ, ένα πρόγραμμα υπολογιστή ή και έναν απλό αισθητήρα. Η περίπτωση του αισθητήρα, π.χ. σε ένα σύστημα θερμοστάτη, είναι το απλούστερο παράδειγμα που μπορεί να δοθεί για έναν πράκτορα χωρίς ευφυΐα, δεδομένου ότι είναι ενσωματωμένος σε ένα φυσικό περιβάλλον και παράγει εξόδους του τύπου «άνοιξε τη βαλβίδα»-«κλείσε τη βαλβίδα». Φυσικά, ένα πιο πολύπλοκο σύστημα αυτοματισμού θα μπορούσε να αναλύει τα δεδομένα που δέχεται ο θερμοστάτης για τη θερμοκρασία του, για να δει το βαθμό συμμετοχής τους στα ασαφή σύνολα της έννοιας «θερμοκρασία» και να λαμβάνει αποφάσεις για το άνοιγμα της βαλβίδας με βάση την ασαφή λογική και λαμβάνοντας επιπλέον υπόψη την αβεβαιότητα που πιθανώς υπάρχει στην ακρίβεια των μετρήσεων. Οι «δαίμονες» λογισμικού που επιτηρούν το περιβάλλον ενός λογισμικού και επεμβαίνουν στη λειτουργία του, για να επιφέρουν μεταβολές, όταν το κρίνουν σκόπιμο, μπορούν να θεωρηθούν πράκτορες. Ελεγκτές εναέριας κυκλοφορίας, συστήματα που παρακολουθούν το ηλεκτρονικό σας ταχυδρομείο και σας ειδοποιούν, όταν υπάρχει νέο μήνυμα, κ.ά. μπορούν να θεωρηθούν πράκτορες.

Συνοπτικά,

ένας πράκτορας αντιλαμβάνεται το περιβάλλον του (μέσω φυσικών αισθητήρων ή αισθητήρων λογισμικού) και δρα, για να το αλλάξει βάσει των στόχων για τους οποίους έχει σχεδιαστεί.

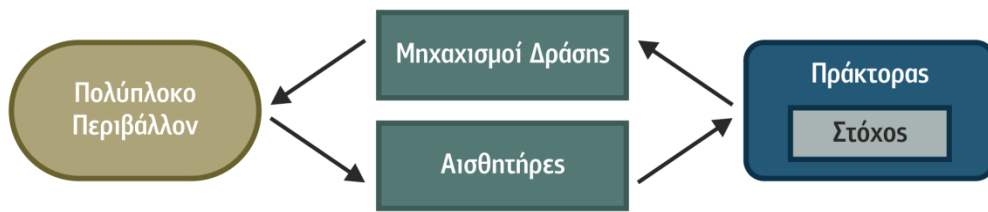
Στη συνέχεια, ακολουθεί μια ανασκόπηση μερικών από τους πολλούς τρόπους με τους οποίους ο όρος «πράκτορας» έχει χρησιμοποιηθεί στο πλαίσιο των αυτόνομων πρακτόρων διακρίνοντάς τους σαφώς από ένα πρόγραμμα. Αυτή η ανασκόπηση θα οδηγήσει σε μια προτεινόμενη ταξινόμηση για τους αυτόνομους πράκτορες.

6.2 Ορισμοί πρακτόρων

Στην εισαγωγή δόθηκε ο βασικός ορισμός των Russel και Norvig (2009) που δίνει έμφαση στην αλληλεπίδραση πρακτόρων με το περιβάλλον τους. Παρά ταύτα, υπάρχει ένα πλήθος ορισμών οι οποίοι προσπαθούν να καλύψουν τον τομέα των πρακτόρων, ο καθένας από διαφορετική σκοπιά. Στην παράγραφο αυτή επιχειρούμε να παρουσιάσουμε το εύρος των ορισμών για τους πράκτορες που συναντώνται συχνά στη βιβλιογραφία, παραθέτοντας τρεις χαρακτηριστικούς από αυτούς.

Η Pattie Maes, του Media Lab στο MIT, είναι από τους πρωτοπόρους της έρευνας στο χώρο των πρακτόρων. Στον ορισμό της για το τι είναι πράκτορας προσθέτει ένα κρίσιμο χαρακτηριστικό: οι πράκτορες πρέπει να δρουν αυτόνομα, ώστε να «συνειδητοποιούν ένα σύνολο στόχων». Το μοντέλο πρακτόρων Maes (Maes, 1995), εκτός της αυτονομίας και της ύπαρξης στόχων, δίνει έμφαση και στο ότι το περιβάλλον είναι πολύπλοκο και πιθανώς δυναμικό:

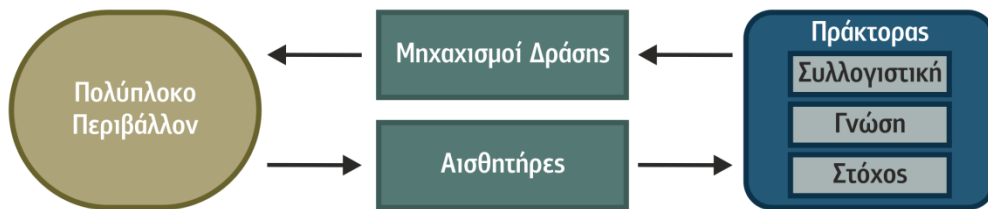
«Οι πράκτορες είναι υπολογιστικά συστήματα που δρουν σε ένα πολύπλοκο περιβάλλον, αντιλαμβάνονται και δρουν αυτόνομα σε αυτό. Με τον τρόπο αυτό επιτυγχάνουν ένα σύνολο από στόχους και εκτελούν καθήκοντα για τα οποία έχουν σχεδιαστεί.»



Σχήμα 6.3 Ο βασικός πράκτορας κατά Maes

Στην περίπτωση κατά την οποία αναφερόμαστε σε ευφυείς πράκτορες, ένας ορισμός που ανταποκρίνεται στα χαρακτηριστικά τους είναι το μοντέλο Hayes-Roth (Hayes-Roth, 1995), που δίνει έμφαση στη συλλογιστική:

«Οι ευφυείς πράκτορες συνεχώς εκτελούν τρεις λειτουργίες: αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος, δρουν στο περιβάλλον, ώστε να το αλλάξουν, και συλλογίζονται, ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα, να εξάγουν συμπεράσματα, για να καθορίσουν τη δράση τους.»



Σχήμα 6.4 Ο βασικός πράκτορας κατά Hayes - Roth

Τέλος, ο SodaBot είναι ένα περιβάλλον ανάπτυξης για πράκτορα λογισμικού που κατασκευάστηκε στο MIT AI Lab από τον Michael Coen το 1994. Ο Coen στο μοντέλο του δίνει έμφαση στη διαδραστικότητα (interactivity):

«Οι λογισμικοί πράκτορες είναι προγράμματα που διενεργούν διάλογο, διαπραγματεύονται και συντονίζουν τη ροή πληροφοριών.»



Σχήμα 6.5 Ο βασικός πράκτορας κατά Coen

6.3 Κοινά Χαρακτηριστικά Πρακτόρων

Βασικό κοινό χαρακτηριστικό των πρακτόρων είναι η αυτονομία τους. Ως αυτόνομες οντότητες ελέγχουν τις καταστάσεις και τις συμπεριφορές τους, χωρίς να κατευθύνονται εξωγενώς βήμα προς βήμα. Άλλα κοινά χαρακτηριστικά που διαθέτουν είναι η κοινωνικότητα και η δυνατότητα συνεργασίας με άλλους πράκτορες:

- *Αυτονομία (autonomy)*: Οι πράκτορες ενεργούν αυτόνομα χωρίς άμεση παρέμβαση από χρήστες ή άλλους πράκτορες, με πλήρη έλεγχο των πράξεών τους (αυτοέλεγχο) και της εσωτερικής τους κατάστασης.
- *Κοινωνικότητα (social ability)*: Οι πράκτορες αλληλεπιδρούν με τους χρήστες για την επίτευξη των στόχων τους και με άλλους πράκτορες μέσω μίας κοινά κατανοητής γλώσσας. Έτσι επιτυγχάνεται επικοινωνία μεταξύ των πρακτόρων για την ολοκλήρωση των ανεξάρτητων στόχων του καθενός ξεχωριστά και ενός κοινού στόχου με συνεργασία μεταξύ τους.
- *Ορθολογικότητα (rationality)*: Αφορά την υπόθεση ότι ένας πράκτορας θα κάνει πάντα το σωστό, δηλαδή θα δρα καταλλήλως για την εκπλήρωση των στόχων του και όχι με τρόπο ο οποίος αποτρέπει την επίτευξή τους.

Οι ευφυείς πράκτορες διαθέτουν επιπλέον χαρακτηριστικά που αφορούν το βαθμό νοημοσύνης που διαθέτουν, όπως:

- **Αντιδραστικότητα (reactiveness)**: Αφορά τον τρόπο με τον οποίο αντιλαμβάνονται το περιβάλλον και ανταποκρίνονται σε τυχόν αλλαγές του εντός συγκεκριμένων χρονικών πλαισίων.
- **Προνοητικότητα (pro-activeness)**: Οι πράκτορες δεν ανταποκρίνονται απλώς στις αλλαγές του περιβάλλοντός τους, αλλά είναι ικανοί να συμπεριφερθούν κατάλληλα σε αυτές τις αλλαγές ορίζοντας επιμέρους στόχους (δηλαδή, αναλαμβάνουν πρωτοβουλία).
- **Γνώση (knowledge)**: Συγκεντρωμένη γνώση σχετική με τον τρόπο με τον οποίο λειτουργεί το περιβάλλον στο οποίο δρα έκαστος πράκτορας, η οποία έχει κατάλληλα αναπαρασταθεί, για να υποστηρίξει τη λήψη αποφάσεων.
- **Πεποιθήσεις (beliefs)**: Αποτελούν την άποψη (view) του πράκτορα για το περιβάλλον του μια δεδομένη χρονική στιγμή, η οποία άποψη ενδέχεται να είναι εσφαλμένη.
- **Επιθυμίες (desires)**: Αφορούν την κρίση του πράκτορα για τις μελλοντικές καταστάσεις του περιβάλλοντός του, όπως, για παράδειγμα, αν μια μελλοντική κατάσταση είναι επιθυμητή ή όχι. Παρά ταύτα, δεν εξετάζεται αν μία επιθυμία του πράκτορα είναι εφικτή ή συγκρούεται με κάποια άλλη.
- **Προθέσεις (intentions)**: Οι προθέσεις είναι υποσύνολο των στόχων, τους οποίους ο πράκτορας προσπαθεί να επιτύχει τη δεδομένη χρονική στιγμή. Δεδομένου ότι δεν είναι δυνατή η ταυτόχρονη επίτευξη όλων των στόχων, επιλέγεται ένα υποσύνολό τους, βάσει ορισμένων κριτηρίων ιεράρχησης.
- **Υποχρεώσεις (obligations)**: Αφορούν την υποχρέωση του πράκτορα να υπακούει σε ένα σύνολο κανόνων και να δρα σε ένα γενικότερο πλαίσιο, ώστε να επιτύχει το σκοπό για τον οποίο σχεδιάστηκε.
- **Προσαρμοστικότητα (adaptivity)**: Ο πράκτορας προσαρμόζεται στο περιβάλλον του (ικανότητα μάθησης).

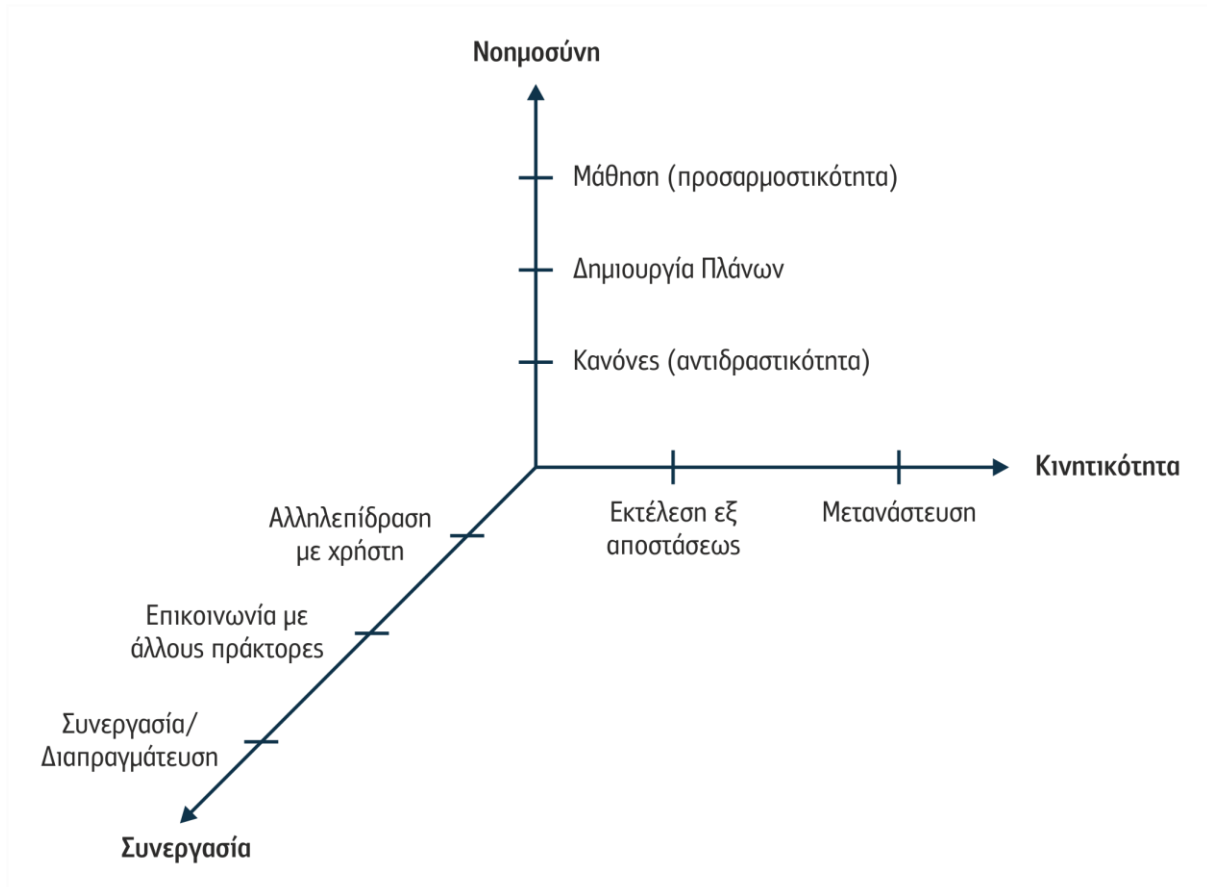
Εκτός των χαρακτηριστικών που προαναφέρθηκαν υπάρχουν και επιπλέον χαρακτηριστικά που απαντούν σε συγκεκριμένες κατηγορίες πρακτόρων προσφέροντας μια «ανθρωπόμορφη αρχιτεκτονική» λογισμικού, που επιτρέπει:

- **Κινητικότητα (mobility)**: Είναι η ικανότητα ενός πράκτορα να μετακινείται ελεύθερα σε ένα φυσικό χώρο ή σε ένα δίκτυο.
- **Συνεργασία (cooperation)** μεταξύ πρακτόρων που βασίζεται σε:
 - **Φιλαλήθεια (veracity)**: Οι πράκτορες δε δίνουν εσκεμμένα λάθος πληροφορίες.

- Αγαθή προαίρεση (benevolence): Ο κάθε πράκτορας προσπαθεί να επιτύχει τους δικούς του στόχους, οι οποίοι βρίσκονται σε αρμονία με τους στόχους των υπολοίπων πρακτόρων του συστήματος.

Τα χαρακτηριστικά των πρακτόρων μπορούν να ομαδοποιηθούν σε τρεις άξονες:

- χαρακτηριστικά που αφορούν γενικά τη συνεργασία των πρακτόρων με τις γύρω τους οντότητες,
- χαρακτηριστικά που αφορούν την κινητικότητα τους,
- χαρακτηριστικά που αφορούν το βαθμό νοημοσύνης τους (βλέπε Σχήμα 6.3).



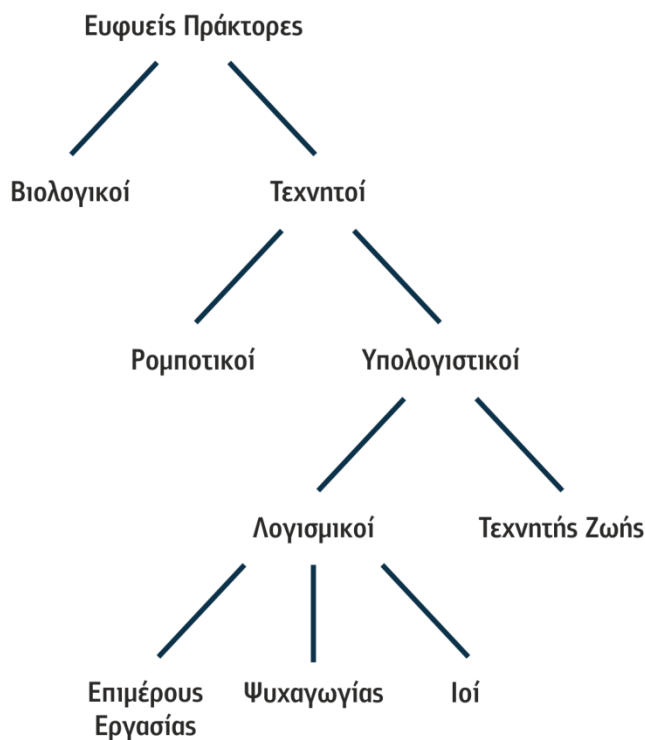
Σχήμα 6.6 Αξονες χαρακτηριστικών πρακτόρων

Στο παραπάνω σχήμα, ο άξονας της συνεργασίας αφορά χαρακτηριστικά αλληλεπίδρασης του πράκτορα με το χρήστη, καθώς και συνεργασία του με άλλους πράκτορες, όταν ανήκει σε ένα πολυ-πρακτορικό σύστημα. Ο άξονας της κινητικότητας αφορά χαρακτηριστικά είτε ρομποτικών πρακτόρων που μετακινούνται, για να δράσουν επιτόπου, είτε λογισμικών που δρουν εξ αποστάσεως. Τέλος, ο άξονας της νοημοσύνης αφορά χαρακτηριστικά που πρέπει να διαθέτει επιπλέον ένας ευφυής πράκτορας, ώστε να είναι ικανός να μαθαίνει, να οργανώνει τις ενέργειές του και να συλλογίζεται, για να αντιδρά «έξυπνα» σε ερεθίσματα που δέχεται.

6.4 Κατηγορίες Πρακτόρων

Παρόλο που οι ευφυείς πράκτορες είναι κατά κάποιον τρόπο νέες εφαρμογές στο χώρο των υπολογιστικών περιβαλλόντων, εμπορικών ή μη, έχουν αποτελέσει αντικείμενο έρευνας από χρόνια. Κατά τη διάρκεια των

χρόνων αυτών έχουν προταθεί αρκετές διαφορετικές κατηγοριοποιήσεις με βάση τους τρεις άξονες των χαρακτηριστικών που προαναφέρθηκαν. Μία από αυτές παρουσιάζεται στο σχήμα που ακολουθεί:



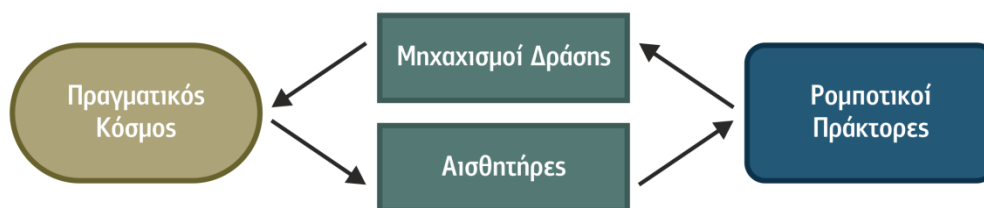
Σχήμα 6.7 Ιεραρχική ταξινόμηση πρακτόρων

Σύμφωνα με την παραπάνω κατηγοριοποίηση, οι ευφυείς πράκτορες χωρίζονται αρχικά σε δυο βασικές κατηγορίες, τους βιολογικούς και τους τεχνητούς.

Οι **βιολογικοί πράκτορες** (biological agents) χρησιμοποιούν τις αισθήσεις τους, για να αντιληφθούν το γύρω κόσμο, τις γνώσεις τους, για να συναγάγουν συμπεράσματα γι' αυτόν, και τα μέρη του σώματός τους, για να εκτελέσουν τις ενέργειες που προκύπτουν από τη συλλογιστική τους.

Οι **τεχνητοί πράκτορες** (artificial agents) λειτουργούν με παρόμοιο τρόπο και χωρίζονται σε δύο υποκατηγορίες, τους ρομποτικούς και τους υπολογιστικούς πράκτορες.

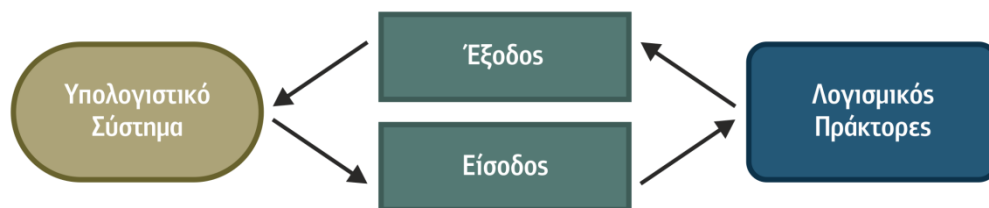
Οι **ρομποτικοί πράκτορες** (robotic agents ή robots) έχουν ως αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο.



Σχήμα 6.8 Λειτουργία ρομποτικού πράκτορα

Στους **υπολογιστικούς πράκτορες** (computational agents) ανήκουν οι **λογισμικοί πράκτορες** (software agents), προγράμματα που λειτουργούν συνεχώς και αυτόνομα στο πλαίσιο ενός υπολογιστικού συστήματος. Ο πράκτορας δραστηριοποιείται με έξυπνο και ευέλικτο τρόπο ανταποκρινόμενος στις αλλαγές του

περιβάλλοντος, χωρίς να απαιτείται η παρέμβαση του χρήστη. Οι λογισμικοί πράκτορες μπορούν να υποδιαιρεθούν σε **πράκτορες επιμέρους εργασιών** (task-specific agents), **πράκτορες ψυχαγωγίας** (entertainment agents) και **ιούς** (viruses). Τέλος, στους υπολογιστικούς πράκτορες ανήκουν οι **πράκτορες τεχνητής ζωής** (artificial life agents) που, επίσης, βασίζονται στο λογισμικό, αλλά έχουν δικά τους ιδιαίτερα χαρακτηριστικά, με βασικότερο ότι λειτουργούν σε εικονικά περιβάλλοντα.



Σχήμα 6.9 Λειτουργία λογισμικού πράκτορα

Σε όποια κατηγορία και αν ανήκει ένας ευφυής πράκτορας, τότε κατά τεκμήριο θα εμπεριέχει τη δυνατότητα λήψης αποφάσεων μέσω κάποιας συλλογιστικής διαδικασίας βασισμένης στην επεξεργασία των ερεθισμάτων που γίνονται δεκτά από το περιβάλλον του. Οι αποφάσεις που λαμβάνονται βάσει της συλλογιστικής διαδικασίας επιδρούν σε αυτό το περιβάλλον, αλλάζοντας με τον τρόπο αυτό την κατάστασή του. Οι λειτουργίες αυτές καθιστούν τους πράκτορες λιγότερο ή περισσότερο ορθολογικούς.

6.4.1 Ορθολογικοί πράκτορες

Οι **ορθολογικοί πράκτορες** (rational agents) λαμβάνουν αποφάσεις και δρουν βάσει μιας διαδικασίας λογικής απόδειξης. Στηρίζονται σε μία βάση γνώσης (knowledge base), χάρη στην οποία διατηρούν την αντίληψή τους για το περιβάλλον με μορφή λογικών προτάσεων (logic formulae), και σε ένα σύνολο κανόνων συμπερασμού, για να δράσουν. Έτσι, οι δράσεις που θα εκτελεστούν ανάγονται σε προβλήματα απόδειξης της μαθηματικής λογικής και πολλές φορές χρησιμοποιούνται συμπερασματικές μηχανές, για να επιτευχθούν.

Οι ορθολογικοί πράκτορες εκτελούν συνεχώς τις εξής τρεις λειτουργίες:

10. αντιλαμβάνονται τις **δυναμικές συνθήκες** του περιβάλλοντος,
11. δρουν στο περιβάλλον, ώστε να **το αλλάξουν** και
12. συλλογίζονται, ώστε να **ερμηνέυσουν** αυτά που αντιλαμβάνονται, να **λύσουν** προβλήματα, να **συμπεράνουν** και να **καθορίσουν** τη δράση τους.

Οι ορθολογικοί πράκτορες χρησιμοποιούν μεθόδους και τεχνικές που έχουν καθορισμένη και απλή σημασιολογία. Συνεπώς, είναι περισσότερο κατάλληλοι για στατικά περιβάλλοντα, ενώ καθίσταται ιδιαίτερα δύσκολη η εξαγωγή συμπερασμάτων σε μεταβαλλόμενα (δυναμικά) περιβάλλοντα, όπου απαιτείται επανασχεδιασμός των δράσεων του πράκτορα και μάλιστα σε περιορισμένα χρονικά όρια. Επίσης, η δράση ενός ορθολογικού πράκτορα σε ένα δυναμικό περιβάλλον απαιτεί ακριβή και ικανοποιητική (αν όχι πλήρη) συμβολική περιγραφή του πραγματικού κόσμου, το οποίο είναι ανέφικτο σε πρακτικό επίπεδο.

6.5 Σύγχρονες Θεωρήσεις Πρακτόρων

Όπως αναφέραμε ήδη, είναι δύσκολη μία ενιαία θεώρηση για το τι είναι ένας πράκτορας, δεδομένου ότι ο κάθε τομέας της Επιστήμης Υπολογιστών τους εξετάζει από διαφορετική σκοπιά. Για παράδειγμα:

- Για την ΤΝ οι πράκτορες είναι ευφυείς οντότητες.
- Για τον αντικειμενοστραφή προγραμματισμό θεωρούνται οντότητες που επικοινωνούν μεταξύ τους.

- Για τον παράλληλο προγραμματισμό είναι οντότητες που εκτελούνται παράλληλα.
- Για την επικοινωνία ανθρώπου-μηχανής είναι οντότητες που αναλαμβάνουν πρωτοβουλία σε συνεργασία με το χρήστη, για να επιτελέσουν τα καθήκοντά τους.

Έτσι προκύπτουν οι εξής δύο σύγχρονες θεωρήσεις για τους πράκτορες, η χαλαρή θεώρηση και η ισχυρή θεώρηση. Η πρώτη ανήκει περισσότερο στο χώρο της τεχνολογίας λογισμικού, ενώ η δεύτερη στο χώρο της ΤΝ. Πιο συγκεκριμένα:

- **Χαλαρή θεώρηση:** ο όρος “πράκτορας” χρησιμοποιείται, για να δηλώσει αυτοπροσδιοριζόμενα υπολογιστικά συστήματα συνήθως, βασισμένα είτε στο λογισμικό (software) είτε στο υλικό (hardware). Κατά τη χαλαρή θεώρηση οι πράκτορες δεν χαρακτηρίζονται απαραίτητα ως “ευφυείς”.
- **Ισχυρή θεώρηση:** στον τομέα της ΤΝ, ένας “πράκτορας”, πέραν των ιδιοτήτων που προαναφέρθηκαν, περιγράφεται με έννοιες και ιδιότητες που αναφέρονται συνήθως σε ανθρώπους, όπως πρόθεση, πεποίθηση, επιθυμία και ικανότητα μάθησης.

6.6 Περιβάλλοντα Πρακτόρων

Μέχρι στιγμής έχει αναφερθεί αρκετές φορές ο όρος “περιβάλλον” μέσα στο οποίο δρα ένας πράκτορας. Γενικότερα, υπό μια έννοια το περιβάλλον αποτελεί το “πρόβλημα” για το οποίο οι πράκτορες αποτελούν τη “λύση”. Οι Russel & Norvig (2009) κατηγοριοποιούν τα περιβάλλοντα των πρακτόρων ως εξής:

- **Πλήρως Προσβάσιμα ή Μερικώς Προσβάσιμα (Fully Observable / Partially Observable):** όταν ένας πράκτορας μπορεί να λαμβάνει μέσω των αισθητήρων του ολοκληρωμένη και ακριβή πληροφορία ανά πάσα στιγμή, τότε το περιβάλλον χαρακτηρίζεται ως προσβάσιμο. Σε ένα πλήρως προσβάσιμο περιβάλλον, ένας πράκτορας δεν απαιτείται να έχει εσωτερική κατάσταση για την αναπαράσταση του περιβάλλοντος. Αντιθέτως, σε ένα μερικώς προσβάσιμο περιβάλλον, οι πράκτορες δε λαμβάνουν άμεση και λεπτομερειακή πληροφορία για αυτό. Σημειωτέον πως ο βαθμός προσβασιμότητας του περιβάλλοντος εξαρτάται και από τους αισθητήρες του εκάστοτε πράκτορα. Για παράδειγμα, σε έναν πράκτορα ο οποίος αναλαμβάνει την παρακολούθηση της θερμοκρασίας του περιβάλλοντος μέσα στο οποίο λειτουργεί μια αντλία και διαθέτει θερμοστάτες μόνο για το νερό το οποίο αντλείται, το περιβάλλον θεωρείται μερικώς προσβάσιμο. Εάν προστεθεί ένας ακόμα θερμοστάτης για το μέταλλο που περιβάλλει την αντλία, τότε το περιβάλλον του πράκτορα θεωρείται πλήρως προσβάσιμο.
- **Αιτιοκρατικά ή Μη Αιτιοκρατικά (Deterministic / Stochastic):** Αν η επόμενη κατάσταση ενός περιβάλλοντος μπορεί να προσδιοριστεί πλήρως από την τρέχουσα κατάστασή του και τις τρέχουσες δράσεις του πράκτορα, τότε αυτό το περιβάλλον καλείται αιτιοκρατικό, ενώ στην αντίθετη περίπτωση χαρακτηρίζεται ως μη αιτιοκρατικό. Μία γενική παρατήρηση είναι ότι ένα περιβάλλον πλήρως προσβάσιμο και αιτιοκρατικό διευκολύνει τον πράκτορα στις δράσεις του. Βέβαια, αν το περιβάλλον είναι μερικώς προσβάσιμο, τότε είναι πολύ πιθανό να αποτελεί παράλληλα ένα μη αιτιοκρατικό περιβάλλον. Αυτό συμβαίνει, κυρίως, σε περιβάλλοντα τα οποία είναι πολύπλοκα και δυσκολεύουν τον πράκτορα στην πλήρη αντίληψη της τρέχουσας κατάστασης του περιβάλλοντός του.
- **Επεισοδιακά ή Μη Επεισοδιακά (Episodic / Sequential):** Ένα περιβάλλον χαρακτηρίζεται ως επεισοδιακό, όταν χωρίζεται σε διακριτά επεισόδια. Κάθε επεισόδιο χαρακτηρίζεται από την αντίληψη του πράκτορα για το περιβάλλον στο οποίο βρίσκεται την τρέχουσα στιγμή και από τις δράσεις του μέσα σε αυτό. Επειδή το κάθε επεισόδιο είναι ανεξάρτητο από δράσεις επεισοδίων που έχουν προηγηθεί, ο πράκτορας δεν εξετάζει την επίδραση που έχουν οι τρέχουσες δράσεις του σε μελλοντικά επεισόδια. Αντίθετα, σε μη επεισοδιακά περιβάλλοντα είναι πιθανό οι τρέχουσες δράσεις να έχουν συνέπειες σε μελλοντικές δράσεις. Για

παράδειγμα, ένας πράκτορας μετεωρολογικών προβλέψεων λειτουργεί σε επεισοδιακό περιβάλλον, ενώ ένας πράκτορας έκδοσης αεροπορικών εισιτηρίων λειτουργεί σε μη επεισοδιακό περιβάλλον.

- **Στατικά** ή **Δυναμικά** (Static / Dynamic): Όταν ένα περιβάλλον μπορεί να μεταβάλλεται από τη φύση του, ενώ παράλληλα δρα μέσα σε αυτό ένας πράκτορας, τότε το περιβάλλον χαρακτηρίζεται ως δυναμικό. Όταν το περιβάλλον μεταβάλλεται, αλλά μόνο λόγω των δράσεων του πράκτορα, καλείται ημιδυναμικό (semi-dynamic). Σε όλες τις άλλες περιπτώσεις χαρακτηρίζεται ως στατικό.
- **Διακριτά** ή **Συνεχή** (Discrete / Continuous): Ο θεωρητικός αυτός διαχωρισμός βασίζεται κυρίως στην αντίληψη του χρόνου από πλευράς του πράκτορα μέσω των αισθητήρων του. Σε ένα διακριτό περιβάλλον, όπως σε ένα παιχνίδι δυο παικτών που παίζεται σε γύρους, οι διάφορες κινήσεις που κάνει κάθε παίκτης στο γύρο του μπορούν να εκφραστούν σε διακριτές τιμές (κινήσεις). Σε ένα συνεχές περιβάλλον, όπως η λειτουργία ενός εργοστασίου, μεταβλητές όπως οι κινήσεις των αυτοματοποιημένων μηχανημάτων αλλάζουν συνεχώς στη μονάδα του χρόνου.
- **Μονοπρακτορικό** ή **Πολυπρακτορικό** (Single Agent / Multiagent): Αν ένας πράκτορας δρα μόνος του σε ένα περιβάλλον, το περιβάλλον χαρακτηρίζεται περιβάλλον μοναδικού πράκτορα, ενώ σε αντίθετη περίπτωση το περιβάλλον είναι πολυπρακτορικό.

Οι παραπάνω κατηγοριοποιήσεις περιγράφουν εντέλει την πολυπλοκότητα ενός περιβάλλοντος. Όσο πιο πολύπλοκο είναι το περιβάλλον, τόσο πιο δύσκολη είναι η ανάπτυξη ενός λειτουργικού πράκτορα που δρα μέσα σε αυτό. Μία σημαντική παρατήρηση είναι ότι η πολυπλοκότητα δεν είναι μία προκαθορισμένη ιδιότητα του περιβάλλοντος, αλλά εξαρτάται σε μεγάλο βαθμό από τους αισθητήρες του πράκτορα που αλληλεπιδρά με αυτό. Δύο πράκτορες ίδιας λειτουργίας αλλά διαφορετικών αισθητήρων οι οποίοι δρουν στο ίδιο περιβάλλον το αντιλαμβάνονται διαφορετικά.

6.7 Αφηρημένες Αρχιτεκτονικές Πρακτόρων

Στις προηγούμενες ενότητες, αναφέρθηκαν αρκετοί ορισμοί των πρακτόρων και τα χαρακτηριστικά ενός πράκτορα, ώστε να δοθεί μια γενική εικόνα της οντότητας των πρακτόρων. Όμως, τίθεται ένα εύλογο ερώτημα: πώς ένας πράκτορας κάνει εκτίμηση των αλλαγών του περιβάλλοντος στο οποίο βρίσκεται και πώς ενεργεί, ώστε να εκπληρώσει επιτυχώς τους στόχους του;

Σε αυτό το ερώτημα καλείται να απαντήσει η αφηρημένη αρχιτεκτονική πρακτόρων, η οποία εξετάζει γενικά την δομή των πρακτόρων, αναλύει τους πράκτορες στα επιμέρους συστατικά στοιχεία τους και περιγράφει τη ροή της πληροφορίας μεταξύ αυτών.

Η αρχιτεκτονική ενός πράκτορα βασίζεται

- στα επιθυμητά χαρακτηριστικά του πράκτορα,
- στα γενικά και ειδικά χαρακτηριστικά του περιβάλλοντος μέσα στο οποίο λειτουργεί ο πράκτορας.

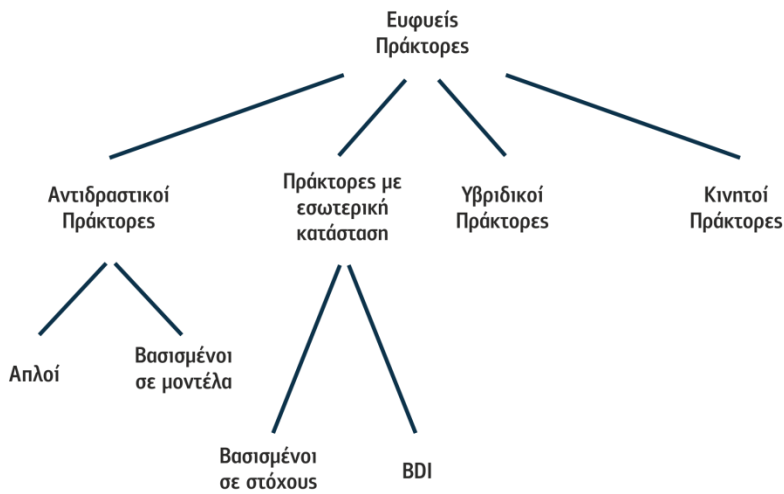
Σκοπός είναι να ορίζει:

- τα βασικά στοιχεία (modules) από τα οποία αποτελείται,
- τον τρόπο με τον οποίο αυτά διασυνδέονται,
- τη ροή της πληροφορίας ανάμεσα στα βασικά στοιχεία.

Με βάση την αφηρημένη αρχιτεκτονική τους, οι πράκτορες μπορεί να κατηγοριοποιηθούν σε δύο βασικές αρχιτεκτονικές ευφυών πρακτόρων, οι οποίες είναι οι **αντιδραστικοί πράκτορες** (reactive agents) και οι **πράκτορες με εσωτερική κατάσταση** (internal state agents). Η βασική διαφορά των δύο αρχιτεκτονικών έγκειται στο ότι χαρακτηρίζονται η μεν πρώτη κατηγορία από τη λογική «ερέθισμα-

αντίδραση», η δε δεύτερη από την ύπαρξη εσωτερικής συμβολικής αναπαράστασης του περιβάλλοντος κόσμου.

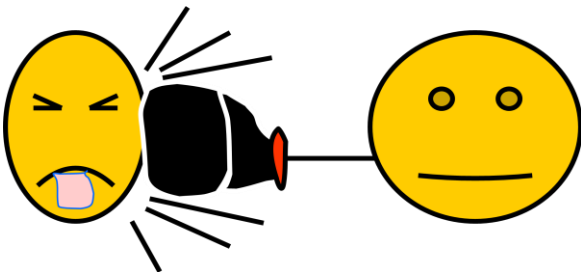
Επιπλέον κατηγορίες είναι οι **υβριδικοί πράκτορες** με διαστρωματωμένη αρχιτεκτονική που δανείζεται από τις δυο βασικές κατηγορίες, και οι **κινητοί πράκτορες**, με χαρακτηριστικό τους το ότι κινούνται ελεύθερα στο περιβάλλον τους.



Σχήμα 6.10 Κατηγορίες ευφυών πρακτόρων βάσει της αρχιτεκτονικής τους

6.7.1 Αντιδραστικοί Πράκτορες

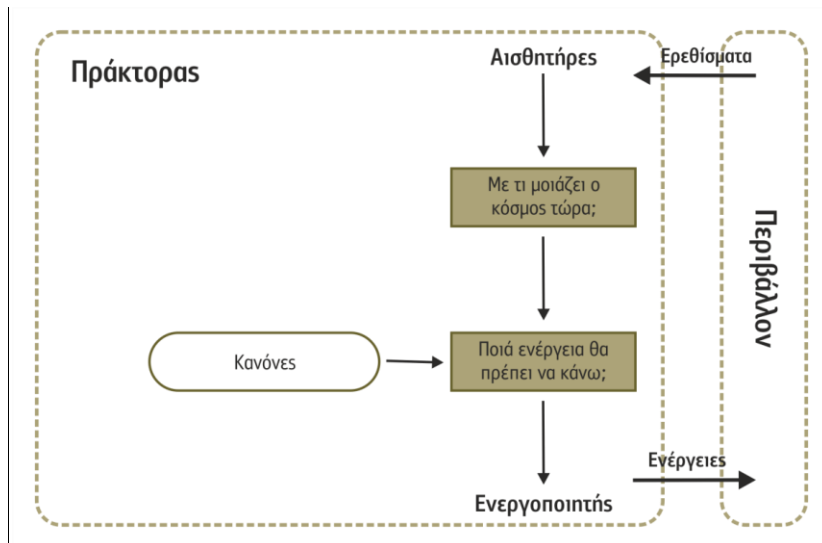
Η αρχιτεκτονική ενός **αντιδραστικού πράκτορα** (reactive agents) αποσκοπεί στη δημιουργία μιας αρχιτεκτονικής όσο γίνεται πιο απλής. Σε αντίθεση με τους πράκτορες με εσωτερική κατάσταση, οι αντιδραστικοί πράκτορες δεν χαρακτηρίζονται από μία εσωτερική συμβολική αναπαράσταση του περιβάλλοντος με το οποίο αλληλεπιδρούν, αλλά η συμπεριφορά τους βασίζεται στη λογική *ερέθισμα-αντίδραση* στην τρέχουσα κατάσταση του κόσμου όπου βρίσκονται. Επιπλέον, δεν έχουν μνήμη. Υπάρχουν τρεις τύποι αντιδραστικών πρακτόρων: οι απλοί, αυτοί που βασίζονται σε μοντέλα του κόσμου τους και αυτοί που βασίζονται σε στόχους.



Εικόνα 6.1 Συμπεριφορά απλού αντιδραστικού πράκτορα

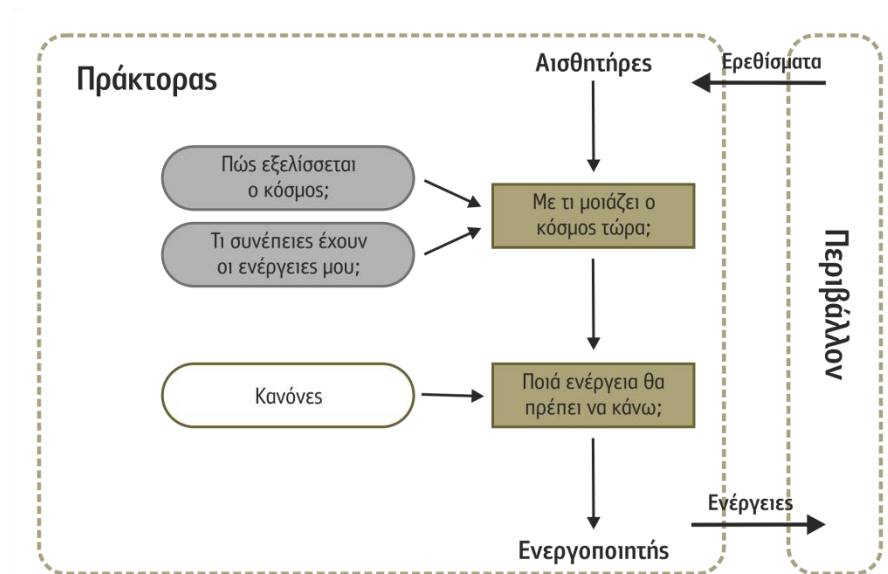
Ένας αντιδραστικός πράκτορας προσδιορίζεται πλήρως από ένα μηχανισμό επιλογής της εκάστοτε κατάλληλης δράσης, που εκτελείται από το τμήμα εκείνο του πράκτορα που καλείται **ενεργοποιητής** (activator).

Ο **απλός αντιδραστικός πράκτορας** (simple reactive agent) διαθέτει μια τρέχουσα αντίληψη για το περιβάλλον του αγνοώντας τι έχει προηγηθεί και χωρίς να τον απασχολεί τι θα ακολουθήσει, αφού δράσει. Ο μηχανισμός αυτός μπορεί να είναι μια συνάρτηση γραμμένη σε κάποια γλώσσα προγραμματισμού ή η μηχανή εξαγωγής συμπερασμάτων σε ένα έμπειρο σύστημα.



Σχήμα 6.11 Αρχιτεκτονική απλού αντιδραστικού πράκτορα

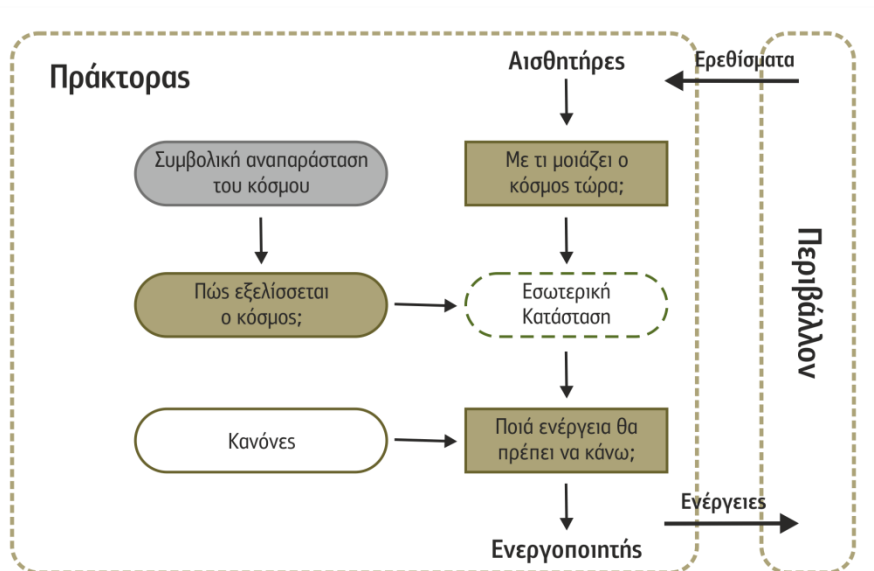
Ο **αντιδραστικός πράκτορας βασισμένος σε μοντέλο** (model-based reactive agent) αντιλαμβάνεται την τρέχουσα κατάσταση του κόσμου μέσω των αισθητήρων του και τη συγκρίνει με ένα εσωτερικό μοντέλο που διαθέτει για το πώς εξελίσσεται ο κόσμος του. Στη συνέχεια βρίσκει έναν κατάλληλο κανόνα με βάση τον οποίο ενεργεί όπως και ο απλός αντιδραστικός πράκτορας.



Σχήμα 6.12 Αρχιτεκτονική αντιδραστικού πράκτορα βασισμένου σε μοντέλα

6.7.2 Πράκτορες με Εσωτερική Κατάσταση

Στην κατηγορία **πρακτόρων με εσωτερική κατάσταση** ανήκουν οι πράκτορες οι οποίοι έχουν μία συμβολική αναπαράσταση του κόσμου μέσα στον οποίο βρίσκονται και διαθέτουν ένα σύνολο κανόνων συμπερασμού βάσει των οποίων σχεδιάζουν την επόμενη ενέργειά τους και μελετούν τα αποτελέσματά της στην εσωτερική αναπαράσταση. Στην προσπάθειά τους να επιτύχουν τους στόχους τους, χρησιμοποιούν κλασικές συλλογιστικές μεθόδους και τεχνικές ταυτοποίησης προτύπων και χειρισμού συμβόλων.



Σχήμα 6.13 Γενικό μοντέλο πράκτορα με εσωτερική κατάσταση

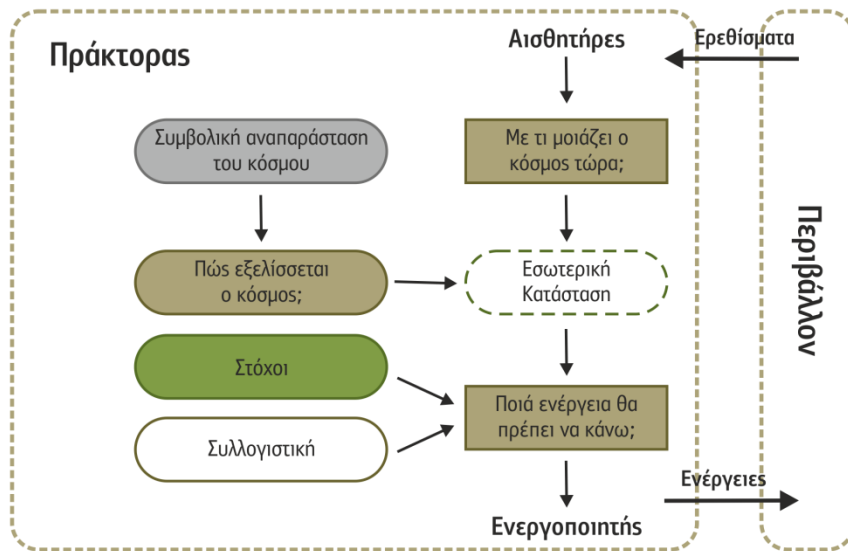
Όταν οι πράκτορες εσωτερικής κατάστασης έχουν μια βάση γνώσης, στην οποία διατηρούν την αντίληψή τους για τον πραγματικό κόσμο με μορφή λογικών προτάσεων και ένα σύνολο συμπερασματικών κανόνων που μπορεί να εκτελέσει ο πράκτορας υπό τύπον συλλογιστικής, τότε καλούνται **πράκτορες βασισμένοι στη λογική** (deliberated agents).

Η κατηγορία των πρακτόρων με εσωτερική κατάσταση περιλαμβάνει δύο υποκατηγορίες αρχιτεκτονικής πρακτόρων: τους **πράκτορες με στόχους**, που είναι βασισμένοι στους επιδιωκόμενους στόχους και τους **πράκτορες με πεποιθήσεις - επιθυμίες - προθέσεις** (βλέπε [Σχήμα 6.10](#)).

Πράκτορες με στόχους

Ένας **πράκτορας με στόχο** (goal-based agent), εκτός των πληροφοριών που προσλαμβάνει μέσω των αισθητήρων του για την τρέχουσα κατάσταση του κόσμου, διαθέτει και πληροφορίες που περιγράφουν τον επιδιωκόμενο στόχο. Ένας **στόχος** (goal) είναι ένα υποσύνολο επιθυμητών μελλοντικών καταστάσεων οι οποίες μπορούν να πραγματοποιηθούν και δεν περιέχουν συγκρούσεις μεταξύ τους. Ο πράκτορας συνδυάζει αυτές τις πληροφορίες με τη γνώση που έχει εσωτερικά για τον κόσμο, ώστε να διερευνήσει τα αποτελέσματα των δυνατών ενεργειών του, προκειμένου να επιλέξει την καταλληλότερη ενέργεια που θα οδηγήσει στο στόχο.

Η εσωτερική κατάσταση μπορεί να βασίζεται σε μοντέλα ή σε οποιαδήποτε άλλη μορφή αναπαράστασης γνώσης.



Σχήμα 6.14 Αρχιτεκτονική πρακτόρων με εσωτερική κατάσταση και στόχο

Πράκτορες με Πεποιθήσεις– Επιθυμίες– Προθέσεις

Ο τρόπος λειτουργίας ενός **πράκτορα BDI** (BDI agent) στηρίζεται σε θεωρητικά μοντέλα ανθρώπινης συλλογιστικής (Weiss, 1999). Βάσει αυτών, η εσωτερική του κατάσταση περιγράφεται και από τις εξής βασικές έννοιες της ισχυρής θεωρήσης των πρακτόρων: “πεποιθήσεις”, “επιθυμίες” και “προθέσεις”.

- Οι **πεποιθήσεις** (beliefs) αντιπροσωπεύουν την αντίληψη του πράκτορα για τον κόσμο· με άλλα λόγια, τις πεποιθήσεις του για τον κόσμο (συμπεριλαμβανομένου και του ίδιου και άλλων πρακτόρων). Οι πεποιθήσεις μπορούν, επίσης, να περιλαμβάνουν συμπερασματικούς κανόνες, επιτρέποντας την προς τα εμπρός αλυσιδωτή εκτέλεσή τους (βλέπε [κεφάλαιο 3](#)), για να οδηγηθούν σε νέες πεποιθήσεις. Η χρήση του όρου πεποίθηση και όχι γνώση αναγνωρίζει ότι αυτό που ένας πράκτορας πιστεύει μπορεί να μην είναι ορθό (και, στην πραγματικότητα, μπορεί να αλλάξει στο μέλλον).
- Οι **επιθυμίες** (desires) αντιπροσωπεύουν την κινητήρια δύναμη του πράκτορα. Είναι στόχοι ή καταστάσεις που ο πράκτορας θα ήθελε να επιτύχει ή να επιφέρει. Για παράδειγμα, επιθυμίες μπορεί να είναι: βρες την οικονομικότερη πτήση προς ένα προορισμό, επένδυσε στην καλύτερη μετοχή ή βρες όλες τις σχετικότερες αναφορές για τον όρο.
 - Στόχοι: Στόχος είναι μια επιθυμία που έχει υιοθετηθεί ως επόμενη επιδίωξη από τον πράκτορα.
- Οι **προθέσεις** (intentions) αντιπροσωπεύουν το τι ο πράκτορας έχει επιλέξει να κάνει. Είναι επιθυμίες στις οποίες ο πράκτορας έχει σε κάποιο βαθμό αφιερωθεί. Πρακτικά, αυτό σημαίνει ότι ο παράγοντας έχει αρχίσει την εκτέλεση ενός προγράμματος.
 - Τα **πλάνα** (plans) είναι ακολουθίες ενεργειών που μπορεί ο πράκτορας να εκτελέσει για την επίτευξη μιας ή περισσότερων από τις προθέσεις του. Τα πλάνα μπορεί να περιλαμβάνουν άλλα πλάνα: σε ένα παιχνίδι, το πλάνο να επισκεφτεί ο ήρωας έναν άλλο πλανήτη περιέχει το πλάνο να σχεδιάσει τη διαδρομή έως εκεί. Αυτό σημαίνει ότι αρχικά τα πλάνα έχουν ένα γενικό σχεδιασμό και, στη συνέχεια, συμπληρώνονται με στοιχεία ανάλογα με το πώς εξελίσσονται.

Η λειτουργία ενός πράκτορα BDI βασίζεται στην ύπαρξη συμβάντων που προκαλούν την αντίδρασή του:

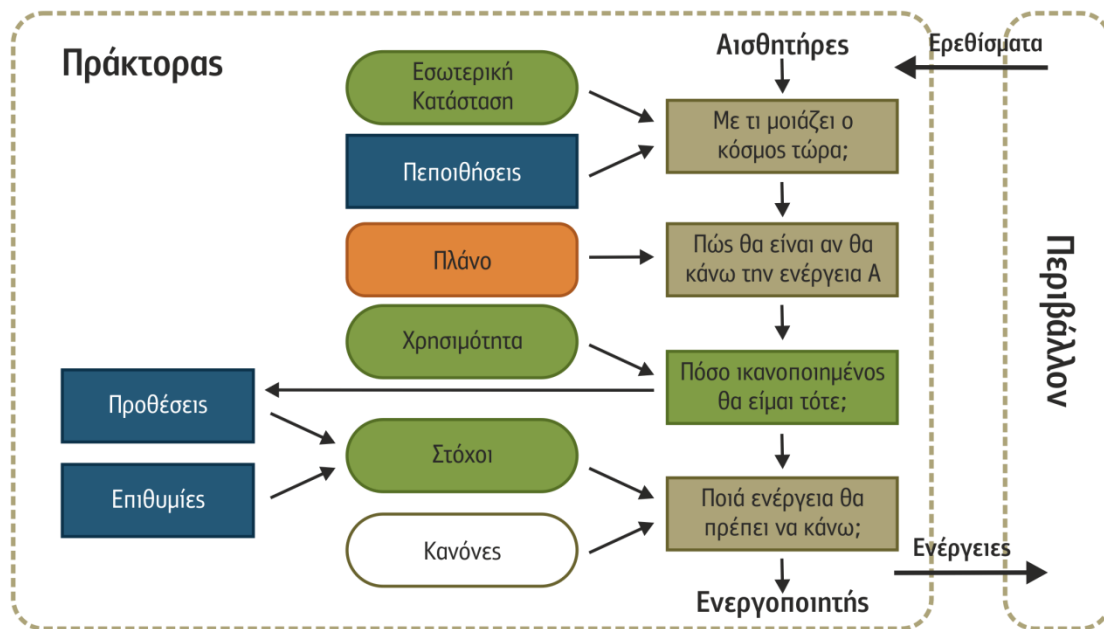
- Τα **συμβάντα** (events) είναι ωθήσεις για αντιδραστική δραστηριότητα από τον πράκτορα. Ένα συμβάν μπορεί να ενημερώσει τις πεποιθήσεις, να προκαλέσει σχέδια ή να τροποποιήσει τους στόχους. Τα συμβάντα μπορούν να δημιουργηθούν είτε εσωτερικά από τον ίδιο τον πράκτορα, για να προκαλέσουν αποσυνδεδεμένες ενημερώσεις ή σχέδια δράσης, ή από το ίδιο το σύστημα είτε εξωτερικά και να γίνουν αντιληπτά από τους αισθητήρες.

Η διαδικασία δράσης ενός πράκτορα BDI, με βάση όλα όσα προαναφέρθηκαν, είναι η ακόλουθη:

1. Ο πράκτορας, μέσα από ένα σύνολο *επιθυμιών*, επιλέγει ανάλογα με την κρίση του τους στόχους του.
2. Στη συνέχεια επιλέγει ένα υποσύνολο από αυτούς (προθέσεις) με βάση το πόσο τον ικανοποιούν (χρησιμότητα) και αφοσιώνεται σε αυτούς.
3. Στο τελικό στάδιο της διαδικασίας δράσης αναλύει τους τρόπους (πλάνα) με τους οποίους οι επιλεγμένες προθέσεις μπορούν να επιτευχθούν.

Η ικανοποίηση του πράκτορα από την επιλογή κάποιου στόχου καλείται **χρησιμότητα** (utility). Η χρησιμότητα προκύπτει από τη **συνάρτηση χρησιμότητας** (utility function) που αντιστοιχίζει μια κατάσταση σε έναν πραγματικό αριθμό που καλείται βαθμός ικανοποίησης. Οι επιλογές που ικανοποιούν τον πράκτορα γίνονται *προθέσεις*, οι οποίες στη συνέχεια προσδιορίζουν τις ενέργειες του πράκτορα.

Οι *προθέσεις* επιδρούν στις *πεποιθήσεις*, πάνω στις οποίες θα βασιστεί η μελλοντική συλλογιστική.



Σχήμα 6.15 Η Αρχιτεκτονική BDI

Αξίζει να αναφερθεί ότι ένας πράκτορας BDI, αφού επιλέξει τους στόχους του από το σύνολο των προθέσεων του, δεσμεύεται πλέον από αυτούς στις μελλοντικές του ενέργειες. Σημειωτέα η επιμονή του πράκτορα προς την επίτευξη των προθέσεων. Αφού έχει αποφασίσει ότι μία πρόθεση είναι εφικτή, ο πράκτορας αφιερώνει χρόνο επεξεργασίας και υπολογιστικούς πόρους, για να την πετύχει. Βέβαια, αν κριθεί ότι αυτές οι προθέσεις είναι αδύνατον τελικά να επιτευχθούν ή σε τελικό στάδιο δεν ικανοποιούν τον πράκτορα στην λειτουργία που θέλει να επιτελέσει, οι προθέσεις αυτές εγκαταλείπονται. Έτσι, ο πράκτορας θα επανεξετάσει τις προθέσεις του για την εύρεση πιο κατάλληλων στόχων. Η συχνότητα επανεξέτασης των προθέσεων του πράκτορα αποτελεί σημαντικότατο κομμάτι στον σχεδιασμό του και επηρεάζει σε μεγάλο βαθμό την αποδοτικότητά του.

Ένα παράδειγμα που βοηθά στην κατανόηση των παραπάνω είναι το ηλεκτρονικό παιχνίδι όπου ένας παίχτης κυνηγάει να σκοτώσει έναν εξωγήινο σε έναν ανοιχτό χώρο στον οποίο κινούνται και οι δυο.

- Στο χρόνο $t = 0$ ο παίχτης έχει ως:
 - επιθυμία να σκοτώσει τον εξωγήινο,
 - πρόθεση να φτάσει στο σημείο P που έχει πληροφορηθεί ότι βρίσκεται ο εξωγήινος,
 - πεποίθηση ότι ο εξωγήινος βρίσκεται στο σημείο P.
- Στον επόμενο χρόνο $t = 1$ ο παίχτης μεταφέρεται στο σημείο P και έχει ως:
 - επιθυμία να σκοτώσει τον εξωγήινο (παραμένει),
 - πρόθεση να σκοτώσει τον εξωγήινο (δεδομένου ότι πιστεύει ότι τον έχει φτάσει),
 - πεποίθηση ότι ο εξωγήινος είναι στο σημείο P (παραμένει).

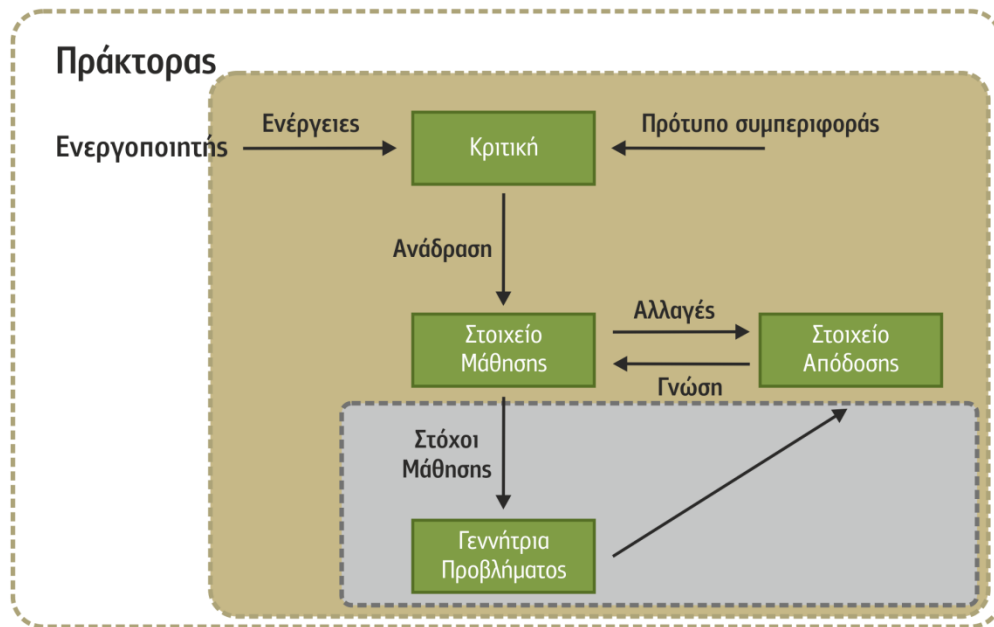
Δεδομένου όμως ότι ο εξωγήινος επίσης κινείται, η πρόθεση του παίχτη δε μπορεί να επιτευχθεί λόγω λανθασμένης πεποίθησης άρα οφείλει να την επανεξετάσει.

Πράκτορες που μαθαίνουν

Οι πράκτορες μπορούν να χαρακτηριστούν πραγματικά ευφυείς, όταν είναι σε θέση να μαθαίνουν, για να βελτιώσουν την απόδοσή τους. Πρακτικά, αυτό σημαίνει ότι η αφηρημένη αρχιτεκτονική τους έχει ενσωματωμένο ένα τμήμα μάθησης ικανό να:

- Κρίνει τη συμπεριφορά του πράκτορα βάσει ενός προτύπου συμπεριφοράς.
- Παρέχει προτάσεις ανάδρασης βάσει της κριτικής που έχει ασκηθεί σε έναν ενσωματωμένο μηχανισμό μηχανικής μάθησης.
- Δημιουργεί αλλαγές στη συμπεριφορά του πράκτορα με τη βοήθεια του μηχανισμού μηχανικής μάθησης.
- Μελετά την απόδοση των αλλαγών και επιστρέφει τις αποδεκτές αλλαγές ως νέα γνώση του πράκτορα, συνηθέστερα σε μορφή νέων ή βελτιωμένων κανόνων.

Ένα πιο πλούσιο μοντέλο πράκτορα που μαθαίνει θα μπορούσε να θέτει επιπλέον στόχους μάθησης για προβλήματα που δεν έχει ακόμα συναντήσει και για τα οποία δεν μπορεί, επομένως, να κρίνει και να βελτιώσει τη συμπεριφορά του. Για την περίπτωση αυτή, είναι απαραίτητη μια γεννήτρια προβλημάτων που θα αντικαθιστά τους αισθητήρες με την εισαγωγή εικονικών ερεθισμάτων για τα οποία θα μελετά την απόδοση του πράκτορα.

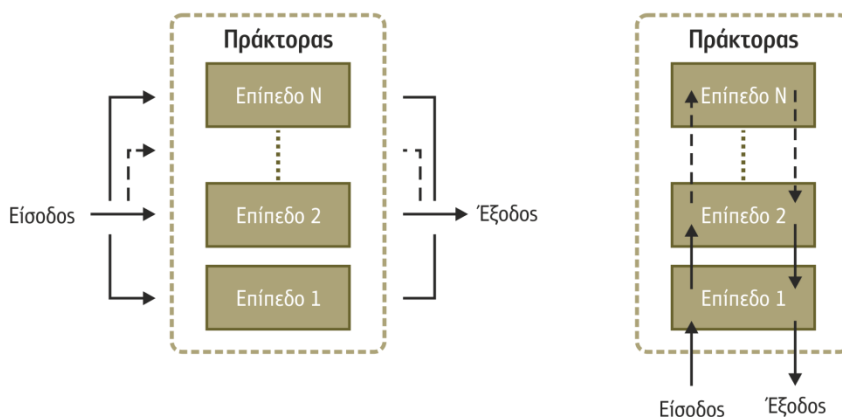


Σχήμα 6.16 Αποτύπωση διαδικασιών μάθησης αντιδραστικών πρακτόρων με μοντέλο

6.7.3 Υβριδικοί Πράκτορες

Η αρχιτεκτονική των **υβριδικών πρακτόρων** (hybrid agents) βασίζεται στο συνδυασμό των δύο προηγούμενων αρχιτεκτονικών με σκοπό την εκμετάλλευση των πλεονεκτημάτων τους. Στους υβριδικούς πρακτορες υπάρχουν τουλάχιστον δύο επίπεδα υπεύθυνα για την αντιδραστική συμπεριφορά και την συμπεριφορά με εσωτερική κατάσταση.

Η ροή ελέγχου σε αυτά τα συστήματα εμφανίζεται σε δύο μοντέλα, το οριζόντιο και το κάθετο (βλέπε Σχήμα 6.17). Στο οριζόντιο μοντέλο, όλα τα επίπεδα είναι συνδεδεμένα σε παραλληλία στους αισθητήρες εισόδου και στους μηχανισμούς δράσης. Στο κάθετο μοντέλο, τα επίπεδα είναι συνδεδεμένα σε σειρά, όπου μόνο ένα επίπεδο είναι συνδεδεμένο στους αισθητήρες εισόδου και μόνο ένα στους μηχανισμούς δράσης.



Σχήμα 6.17 Ροή ελέγχου σε υβριδικούς πρακτορες

6.7.4 Κινητοί Πράκτορες

Η τελευταία κατηγορία λογισμικών πρακτόρων που θα εξετάσουμε είναι οι **κινητοί πράκτορες** (mobile agents). Ιδιαίτερο χαρακτηριστικό της κατηγορίας αυτής είναι η ικανότητα των πρακτόρων να μετακινούνται μέσα στο περιβάλλον τους που συνήθως είναι διαδικτυακό. Οι κινητοί πράκτορες είναι στην πραγματικότητα διεργασίες λογισμικού (software processes), οι οποίες έχουν την δυνατότητα να εκτελούνται σε οποιονδήποτε υπολογιστή και αν μεταφερθούν.

Ο τρόπος λειτουργίας των κινητών πρακτόρων δεν απαιτεί τη συνεχή επικοινωνία μεταξύ του εξυπηρετητή (server) όπου είναι εγκατεστημένος ο πράκτορας και του υπολογιστή στον οποίο μεταφέρεται, για να δράσει. Με αυτό τον τρόπο υπάρχει ελάχιστη κατανάλωση στους πόρους του δικτύου και η ακεραιότητα των πληροφοριών που λαμβάνει ο πράκτορας δεν επηρεάζεται από τυχόν προβλήματα επικοινωνίας με τον εξυπηρετητή. Για παράδειγμα, ένας κινητός πράκτορας που πρέπει να συλλέξει πληροφορίες από διάφορους ιστοχώρους θα πρέπει να μετακινηθεί στους εξυπηρετητές των ιστοχώρων αυτών στη σειρά, δηλαδή να επισκεφτεί τον πρώτο στη σειρά, όπου θα αντλήσει τις πληροφορίες που τον ενδιαφέρουν, στη συνέχεια θα μεταβεί στο δεύτερο και ούτω καθ' εξής, μέχρι να επιστρέψει στη βάση του επιβαρύνοντας μόνο το σύστημα του υπολογιστή στον οποίο ενεργεί κάθε χρονική στιγμή. Αντίθετα, ένας μη κινητός πράκτορας θα στείλει αιτήματα προς τους εξυπηρετητές των ιστοσελίδων με τα στοιχεία που τον ενδιαφέρουν και, αφού επεξεργαστεί τις απαντήσεις που θα λάβει, θα παρουσιάσει τα αποτελέσματα στο χρήστη, διαδικασία που αντιστοιχεί σε μεγάλο επικοινωνιακό κόστος και όγκο μεταφερόμενων πληροφοριών.

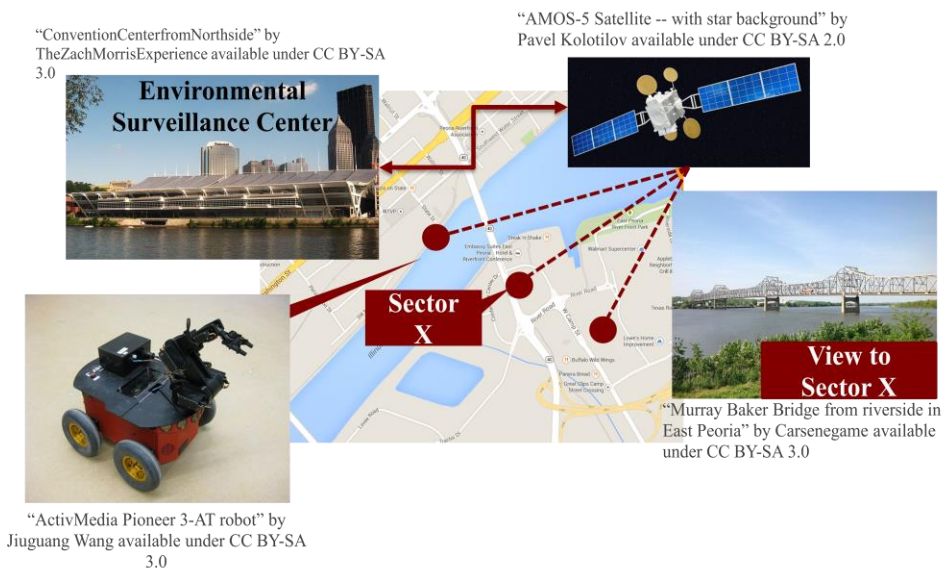
Βέβαια, ο τρόπος λειτουργίας των κινητών πρακτόρων εμπεριέχει μεγάλα κενά ασφάλειας. Κατά την μεταφορά των πρακτόρων μεταξύ των εξυπηρετητών, υπάρχει το ενδεχόμενο αλλοίωσης του κώδικά τους από κακόβουλα λογισμικά, με συνέπεια οι διεργασίες που εκτελούνται να δώσουν πιθανώς μη επιθυμητά αποτελέσματα και, ακόμα χειρότερα, επικίνδυνα για τους ίδιους τους εξυπηρετητές. Επίσης, βρίσκονται σε κίνδυνο και οι υπολογιστές στους οποίους θα μεταφερθούν οι πράκτορες στην συνέχεια, αφού ο λειτουργικός σκοπός τους είναι, πλέον, αμφίβολος. Τέλος, δεδομένου ότι οι πράκτορες μετακινούνται συνέχεια, είναι δύσκολη η παρακολούθηση της τρέχουσας θέσης τους, ώστε να ελεγχθεί η κατάσταση λειτουργίας τους.

6.8 Πολυπρακτορικά Συστήματα

Ένα **πολυπρακτορικό σύστημα** (multi-agent system) αποτελείται από ένα σύνολο από πράκτορες που δρουν μαζί, για να επιλύσουν ένα πρόβλημα. Ένα τέτοιο σύστημα στοχεύει στη διασύνδεση και λειτουργία ήδη υπαρχόντων συστημάτων, καθώς και στην επίλυση προβλημάτων:

- που είναι πέρα των δυνατοτήτων και της γνώσης ενός μόνο πράκτορα
- τα οποία είναι από τη φύση τους κατανεμημένα. Τα πολυπρακτορικά συστήματα αποτελούν βασικό τομέα της Κατανεμημένης ΤΝ από πλευράς χαλαρής θεώρησης των πρακτόρων
- όπου η σχετική γνώση είναι κατανεμημένη σε διακριτές πηγές, όπως για παράδειγμα η υπάρχουσα εμπειρία στα επιμέρους γραφεία ενός οργανισμού.

Παραδείγματα πολυπρακτορικών συστημάτων μπορεί κάποιος να συναντήσει σε διαδικτυακές εφαρμογές παροχής πληροφοριών σε σύνθετα διαδραστικά περιβάλλοντα.



Εικόνα 6.2 Πολυπρακτορικό σύστημα για την περιβαλλοντική εποπτεία, την παρακολούθηση και οικολογική προστασία

6.8.1 Χαρακτηριστικά πολυπρακτορικών συστημάτων

Κύριο χαρακτηριστικό των συνεργαζόμενων πρακτόρων είναι ο συντονισμός (coordination). Για το συντονισμό, ένας αντιπροσωπευτικός ορισμός είναι ο παρακάτω:

«Ο συντονισμός είναι η ιδιότητα ενός συστήματος πρακτόρων να φέρουν εις πέρας ενέργειες μέσα σε ένα κοινό περιβάλλον.»

Υπάρχουν δύο τρόποι συντονισμού:

- Διαπραγμάτευση (Negotiation)
- Συνεργασία (Cooperation)

Άλλα χαρακτηριστικά των πολυπρακτορικών συστημάτων είναι τα ακόλουθα:

- Κανένας πράκτορας δεν έχει πλήρη πληροφορία γιατί τα δεδομένα είναι κατανεμημένα.
- Δεν υπάρχει κεντρικός έλεγχος στο σύστημα.
- Οι υπολογισμοί γίνονται με ασύγχρονο τρόπο.

Τα χαρακτηριστικά ενός πολυπρακτορικού συστήματος επιδρούν παράλληλα στον σχεδιασμό και τη λειτουργία των πρακτόρων που το αποτελούν.

6.8.2 Κρίσιμα σημεία στη σχεδίαση & υλοποίηση

Το σημαντικότερο και πιο κρίσιμο σημείο στην ανάπτυξη πολυπρακτορικών συστημάτων είναι ο τρόπος επικοινωνίας των πρακτόρων, που απαιτεί την ύπαρξη τριών διαφορετικών επιπέδων:

- του κατώτερου επιπέδου (αφορά τον τρόπο διασύνδεσης),
- του μεσαίου επιπέδου (αφορά τη σύνταξη και τη μορφή των μηνυμάτων),

- του ανώτερου επιπέδου (αφορά τη σημασιολογία).

Θέματα που εγείρονται κατά το σχεδιασμό της επικοινωνίας πρακτόρων είναι:

- Ποιες γλώσσες και πρωτόκολλα θα χρησιμοποιηθούν;
- Πότε αυτοί επικοινωνούν και τι πληροφορίες ανταλλάσσουν;
- Με ποιους άλλους πράκτορες επικοινωνούν και ποιο είναι το επιπλέον κόστος;

Η αρμονική συνύπαρξη των μελών ενός πολυπρακτορικού συστήματος θέτει τα θεμέλια για την αποδοτική και σταθερή λειτουργία του. Για την επιτυχή αλληλεπίδραση των πρακτόρων ως μελών μιας ομάδας με κοινό σκοπό, θα απαιτηθεί τυποποίηση, κατάλληλη περιγραφή και διαμοιρασμός του προβλήματος, καθώς και σύνθεση των λύσεων που κάθε πράκτορας μεμονωμένα θα εμφανίζει. Επίσης, προσεκτικό και λεπτομερή σχεδιασμό απαιτούν αφενός ο τρόπος για συμβιβασμό πιθανών διαφορετικών απόψεων μεταξύ πρακτόρων και αντιμετώπιση ενδεχόμενων συγκρουόμενων προθέσεων τους, αφετέρου ο τρόπος διαχείρισης περιορισμένων πόρων.

6.8.3 Μοντέλα διασύνδεσης πολυπρακτορικών συστημάτων

Η αρχιτεκτονική δομή του συστήματος, δηλαδή η διασύνδεση των πρακτόρων του συστήματος μεταξύ τους, απαιτεί σχεδιασμό ανεξάρτητα από την δημιουργία και την ένταξη των πρακτόρων στο σύστημα. Τα επικρατέστερα μοντέλα διασύνδεσης για τη συνεργασία των πρακτόρων είναι τα **συστήματα μαυροπίνακα** (blackboard systems) και τα **συστήματα ανταλλαγής μηνυμάτων** (message passing systems).

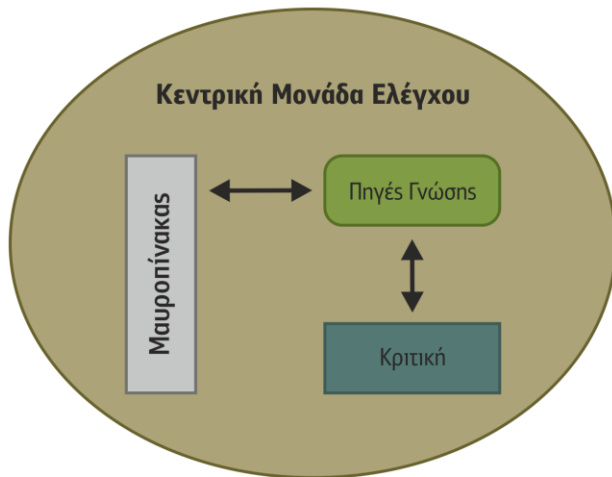
Στο μοντέλο με μαυροπίνακα η διασύνδεση των πρακτόρων γίνεται μέσω μιας κεντρικής μονάδας ελέγχου που καλείται *Μαυροπίνακας* (βλέπε επόμενο σχήμα).



Σχήμα 6.18 Μοντέλο διασύνδεσης Μαυροπίνακα

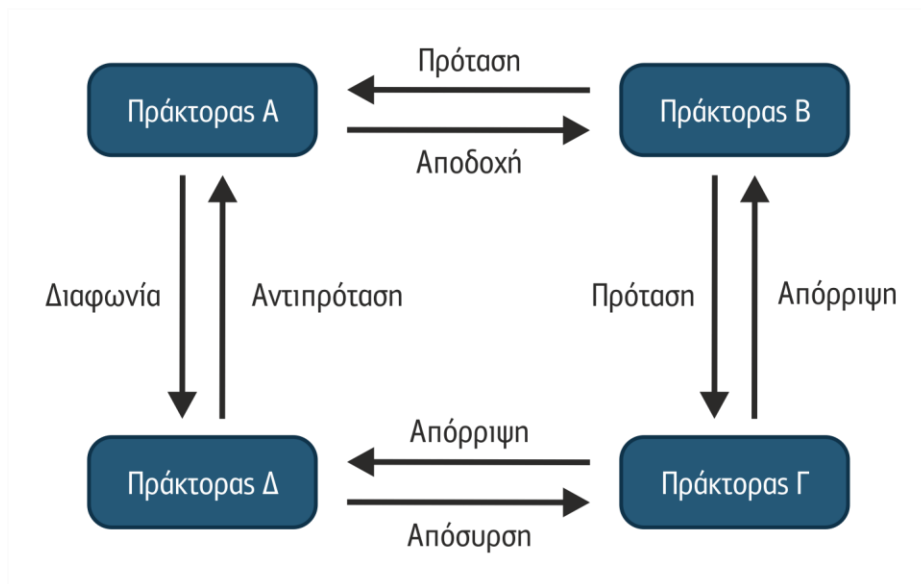
Η κεντρική μονάδα ελέγχου χωρίζεται σε τρία επιμέρους τμήματα, όπως φαίνεται στο παρακάτω σχήμα 6.19:

- **πηγές γνώσης** (knowledge sources), οι οποίες είναι ανεξάρτητα δομικά στοιχεία (modules) τα οποία εμπεριέχουν τη γνώση η οποία χρειάζεται για την επίλυση ενός προβλήματος,
- **στοιχείο ελέγχου** (control component), που κάνει κριτική στις πηγές γνώσης και λαμβάνει αποφάσεις σε πραγματικό χρόνο για την πορεία επίλυσης του προβλήματος και το διαμοιρασμό των πόρων του συστήματος για να την υποστηρίξει,
- **μαυροπίνακα** (blackboard), που είναι μία καθολική βάση δεδομένων αποτελούμενη από δεδομένα εισόδου, μερικές λύσεις και άλλα δεδομένα τα οποία αξιοποιούνται κατά την επίλυση του προβλήματος.



Σχήμα 6.19 Αρχιτεκτονική κεντρικής μονάδας ελέγχου στο μοντέλο διασύνδεσης με Μαυροπίνακα

Στο **μοντέλο ανταλλαγής μηνυμάτων** οι πράκτορες συνεργάζονται για την επίλυση ενός προβλήματος μέσω μηνυμάτων τα οποία αποστέλλουν ο ένας στον άλλο βάσει συγκεκριμένων γλωσσών υψηλού επιπέδου. Σε αντίθεση με το μοντέλο μαυροπίνακα, δεν περιέχεται κάποια κεντρική μονάδα για τον έλεγχο της πορείας επίλυσης προβλημάτων που δημιουργούνται από ασυμφωνίες μεταξύ των πρακτόρων και από ανάγκες διαμοιρασμού των περιορισμένων πόρων του συστήματος. Τα ζητήματα αυτά πρέπει να αναλύονται κατά τον σχεδιασμό του συστήματος ή μέσω ανταλλαγής μηνυμάτων και διαπραγματεύσεων μεταξύ των πρακτόρων. Στο παρακάτω σχήμα παρουσιάζεται η μορφή του συστήματος ανταλλαγής μηνυμάτων σε απλή μορφή.



Σχήμα 6.19 Μοντέλο διασύνδεσης ανταλλαγής μηνυμάτων

Τα χαρακτηριστικά του μοντέλου ανταλλαγής μηνυμάτων είναι:

- Ανταλλάσσεται πληροφορία και υπάρχει συνεργασία μέσω μηνυμάτων.
- Αποστολή μηνυμάτων βάσει συγκεκριμένων γλωσσών υψηλού επιπέδου.
- Επιτρέπει την υλοποίηση πολύπλοκων μοντέλων συνεργασίας μεταξύ των πρακτόρων.

Τα συστήματα που βασίζονται στο μοντέλο ανταλλαγής μηνυμάτων προσφέρουν μεγαλύτερη ευελιξία στην ανταλλαγή πληροφοριών από όσο τα συστήματα μαυροπίνακα.

6.8.4 Επικοινωνία μεταξύ πρακτόρων

Απαραίτητη προϋπόθεση για τη λειτουργία ενός πολυπρακτορικού συστήματος είναι η ύπαρξη πρωτοκόλλων επικοινωνίας και αλληλεπίδρασης. Τα *πρωτόκολλα επικοινωνίας* ορίζουν μηχανισμούς μέσω των οποίων οι πράκτορες μεταδίδουν μεμονωμένα μηνύματα, ενώ τα *πρωτόκολλα αλληλεπίδρασης* κατευθύνουν την ανταλλαγή μίας σειράς μηνυμάτων μεταξύ των πρακτόρων· δηλαδή, πραγματοποιούν διάλογο (Huhns και Stephens, 1999).

Η επικοινωνία μεταξύ των πρακτόρων μπορεί να είναι σύγχρονη ή ασύγχρονη. Κατά τη *σύγχρονη επικοινωνία*, ο πράκτορας που θέτει μία ερώτηση είναι απαραίτητο να διακόψει τη λειτουργία του, μέχρι να πάρει μία απάντηση, ενώ, κατά την *ασύγχρονη*, η απάντηση μπορεί να έλθει οποιαδήποτε στιγμή μετά το χρόνο υποβολής της ερώτησης, χωρίς η λειτουργία του πράκτορα να διακόπτεται.

Επιπρόσθετα, ένα άλλο θέμα που αφορά την επικοινωνία των πρακτόρων σε ένα σύστημα είναι ο *βαθμός επικοινωνίας (cardinality)*. Αυτός ορίζεται ως ο αριθμός των αποστολέων και των παραληπτών κατά την ανταλλαγή πληροφορίας. Ο βαθμός επικοινωνίας μπορεί να είναι:

- 1 προς 1,
- 1 προς N,
- N προς N.

6.8.5 Αλληλεπίδραση Πρακτόρων

Όπως αναφέρθηκε παραπάνω, τα πρωτόκολλα επικοινωνίας καθορίζουν την επικοινωνία μεταξύ των πρακτόρων σε πρωτογενές επίπεδο. Οι πράκτορες, ιδιαίτερα αυτοί με εσωτερική κατάσταση, είναι οντότητες με ξεχωριστούς στόχους, επιθυμίες και πεποιθήσεις και απαιτούν πρωτόκολλα αλληλεπίδρασης που θα τους επιτρέψουν να λειτουργήσουν ως κοινωνία πρακτόρων. Στο πλαίσιο μιας τέτοιας κοινωνίας απαιτείται *συντονισμός (coordination)* των πρακτόρων, να εμποδίζεται η άσχετη με το στόχο δραστηριότητα μέσω της αποφυγής καταστάσεων αδιεξόδου και ενεργούς αναμονής.

Ο συντονισμός σε πολυπρακτορικά περιβάλλοντα μέσα στα οποία οι πράκτορες δεν έχουν αντικρουόμενους στόχους εκφράζεται ως **συνεργασία (cooperation)**, με *πρωτόκολλα συνεργασίας (cooperation protocols)*, ενώ στην αντίθετη περίπτωση ορίζεται ως **διαπραγμάτευση (negotiation)**, με *πρωτόκολλα διαπραγμάτευσης (negotiation protocols)*.

Στα πρωτόκολλα συνεργασίας γίνεται κατάτμηση των διεργασιών στους συνεργαζόμενους πράκτορες. Στα πρωτόκολλα διαπραγμάτευσης βασικός στόχος είναι η από κοινού λήψη αποφάσεων για πράκτορες οι οποίοι έχουν προσωπικούς στόχους και επιθυμίες. Οι πράκτορες αρχικά επικοινωνούν μεταξύ τους για την ανάλυση των στόχων τους και στη συνέχεια διαπραγματεύονται μία αμοιβαία συμφωνία μέσω παραχωρήσεων ή αναζήτησης

Τα πρωτόκολλα αλληλεπίδρασης προσδιορίζουν την τυποποίηση των μηνυμάτων που μπορούν να ανταλλάξουν οι πράκτορες μεταξύ τους. Ένα πρωτόκολλο αλληλεπίδρασης θα μπορούσε να ορίσει την ακόλουθη τυποποίηση των μηνυμάτων που ανταλλάσσουν δύο πράκτορες:

- πρόταση ενέργειας,
- αποδοχή προτεινόμενης ενέργειας,
- απόρριψη προτεινόμενης ενέργειας,
- απόσυρση προτεινόμενης ενέργειας,
- διαφωνία σε μία προτεινόμενη ενέργεια,
- αντιπρόταση σε μία προτεινόμενη ενέργεια.

Βασιζόμενοι σε αυτού του τύπου τα μηνύματα, μία τυπική επικοινωνία μεταξύ του Πράκτορα_A και του Πράκτορα_B, η οποία είναι παράλληλα ένα στιγμιότυπο του πρωτοκόλλου αλληλεπίδρασης για την διαπραγμάτευση, είναι η ακόλουθη:

- Πράκτορας_A: προτείνει ενέργεια σε Πράκτορα_B.
- Πράκτορας_B: αναλύει την πρόταση και εναλλακτικά:
 - Στέλνει μήνυμα αποδοχής στον Πράκτορα_A.
 - Στέλνει μήνυμα αντιπρότασης στον Πράκτορα_A.
 - Στέλνει μήνυμα διαφωνίας στον Πράκτορα_A.
 - Στέλνει μήνυμα απόρριψης στον Πράκτορα_A.

6.8.6 Πολυπρακτορικό περιβάλλον

Η αλληλεπίδραση των πρακτόρων πραγματοποιείται σε ένα πολυπρακτορικό περιβάλλον. Οι πράκτορες σε ένα τέτοιο περιβάλλον μπορεί να έχουν αντικρουόμενους στόχους, όπως για παράδειγμα πράκτορες κρατήσεων θέσεων σε θέατρα, όταν έχουν ως στόχο την κράτηση των τελευταίων περιορισμένων θέσεων μιας θεατρικής παράστασης, ή μπορεί να έχουν αντίστοιχα κοινούς στόχους, όπως για παράδειγμα πράκτορες που συνεργάζονται, για να υποδείξουν στο χρήστη συμφέροντα αεροπορικά εισιτήρια για έναν προορισμό και ταυτόχρονα να του προτείνουν ξενοδοχεία για να μείνει και εταιρείες ενοικίασης αυτοκινήτου. Το πιο χαρακτηριστικό παράδειγμα πολυπρακτορικού περιβάλλοντος είναι το Διαδίκτυο. Πρόκειται για ένα καταναμημένο ανοικτό περιβάλλον συνεχώς εξελισσόμενο με αλλαγές στο μέγεθός του και στα δομικά του στοιχεία, μεταξύ των οποίων είναι και οι πράκτορες.

6.8.7 Γλώσσες επικοινωνίας

Για τη μεταξύ τους επικοινωνία οι πράκτορες χρησιμοποιούν ειδικές γλώσσες επικοινωνίας. Ένα πρωτόκολλο αλληλεπίδρασης ταυτίζεται με τη γλώσσα επικοινωνίας και καθορίζεται από δύο επίπεδα:

- το κατώτερο επίπεδο, το οποίο αφορά τη **σύνταξη** (syntax) ή τη μορφή των μηνυμάτων (format of messages),
- το ανώτερο επίπεδο, το οποίο αφορά τη **σημασιολογία** (semantics) ή το νόημα της πληροφορίας (meaning of information).

Οι κυριότερες γλώσσες επικοινωνίας πρακτόρων από τις αρχές της δεκαετίας του 1990 έως σήμερα είναι η KQML και η FIPA ACL. Η FIPA ACL δημιουργήθηκε από τον Οργανισμό για Ευφυείς Φυσικούς Πράκτορες (Foundation for Intelligent Physical Agents - FIPA) το 1999 και βασίστηκε στην προσπάθεια για διασύνδεση των διαφορετικών συστημάτων πρακτόρων που αναπτύσσονται στη βιομηχανία. Η FIPA ACL μοιάζει αρκετά στην KQML, αλλά έχει μία σημαντική διαφορά. Τα τελεστικά μηνύματα και η σημασιολογία της είναι αυστηρώς ορισμένα.

Η σημασιολογία της FIPA ACL εκφράζεται με τη χρήση της γλώσσας S/SL⁸, η οποία περιλαμβάνει, μεταξύ άλλων, τις πεποιθήσεις, τις επιθυμίες και τις πράξεις των πρακτόρων. Με βάση τη λογική αυτή, κάθε μήνυμα ACL αντιστοιχεί σε ένα ορθολογικό αποτέλεσμα ή, αλλιώς, στο τι αποβλέπει ο αποστολέας του μηνύματος.

Σε γενικές γραμμές η FIPA ACL δεν μπορεί να χρησιμοποιηθεί σε πολυπρακτορικά συστήματα στα οποία δρουν πράκτορες συγκρουόμενων συμφερόντων. Στις περιπτώσεις εκείνες, όμως, κατά τις οποίες έχει διασφαλιστεί ότι δεν ισχύει κάτι τέτοιο, δεν παρουσιάζονται σημαντικά προβλήματα στην χρησιμοποίησή της. Τέλος, αξίζει να σημειωθεί πως το 2005 η FIPA ACL ως δημιούργημα της FIPA εντάχθηκε στην προτυποποίηση του IEEE.

⁸ Η S/SL (Syntax/Semantic Language) δημιουργήθηκε από τους Cordy, Holt και Wortman το 1982 στο Πανεπιστήμιο του Τορόντο. Είναι μία υψηλού επιπέδου γλώσσα προδιαγραφών για σημασιολογικούς αναλυτές, γεννήτριες κώδικα και για μεταγλωττιστές.

6.9 Εφαρμογές πρακτόρων

Οι πράκτορες, ως ευφυή συστήματα, έχουν χρησιμοποιηθεί σε ένα πλήθος βιομηχανικών και εμπορικών εφαρμογών: για παρακολούθηση ασθενών, ρύθμιση εναέριας κυκλοφορίας, σε παιχνίδια, σε προσομοιώσεις, στο ηλεκτρονικό εμπόριο, για υποστήριξη χρηστών σε εφαρμογές γραφείου, κλπ.

Οι πράκτορες διεπαφής αποτελούν προσωπικούς βοηθούς (personal assistants) του χρήστη. Μαθαίνουν τις ιδιαίτερες προτιμήσεις του και τον βοηθούν στην ικανοποίηση αναγκών του · παράδειγμα, ο γνωστός συνδετήρας του προγράμματος Word και τα προγράμματα που βοηθούν τους χρήστες να οργανώσουν το ημερήσιο πρόγραμμά τους (Calendar Agent, Calendar Apprentice).

Κλασικό παράδειγμα πολυπρακτορικού συστήματος είναι αυτό που χρησιμοποιείται για τον **έλεγχο εναέριας κυκλοφορίας**, όπου τα αεροσκάφη και τα διάφορα συστήματα ελέγχου αναπαρίστανται από πράκτορες και κάθε αεροσκάφος που εισέρχεται στην επιχειρησιακή ζώνη του αεροδρομίου ανατίθεται σε έναν διαφορετικό πράκτορα του συστήματος, που αναλαμβάνει την προσγείωση ή απογείωσή του.

Ιδιαίτερη κατηγορία αποτελούν οι πράκτορες διαδικτύου, που διευκολύνουν το χρήστη στην ανεύρεση χρήσιμης πληροφορίας, ενώ προσπαθούν να μάθουν τα ειδικότερα ενδιαφέροντα του χρήστη χρησιμοποιώντας μεθόδους μάθησης μηχανής, για να τον εξυπηρετήσουν καλύτερα. Οι πράκτορες αυτοί ανήκουν κυρίως σε τέσσερις διαφορετικούς τύπους:

- Buyer agent or shopping bot – βοήθεια προς τον πελάτη να βρει προϊόντα και υπηρεσίες.
 - Το MySimon το επιτυχέστερο bot έως σήμερα
- User or personal agents – προσωποποιημένη υποστήριξη χρήστη
- Monitoring-and-surveillance agents (καλούνται επίσης predictive agents) – παρατήρηση άλλων συστημάτων και αναφορά στον ίδιο τον εξοπλισμό που τα υποστηρίζει.
- Data-mining agents – ενεργούν σε αποθηκευμένα δεδομένα για την εξόρυξη χρήσιμων δεδομένων.

Τέλος, η διαθεσιμότητα του παγκόσμιου χώρου των δεδομένων (Web of Data) δημιουργεί νέες ευκαιρίες για σχεδιασμό ευφυών πρακτόρων που θα εκμεταλλεύονται τεχνικές της ΤΝ σε σχέση με την αναπαράσταση γνώσης, την ανάκτηση πληροφορίας, την εξόρυξη δεδομένων και την ενσωμάτωση πληροφοριών σε βάσεις δεδομένων.

Βιβλιογραφία/Αναφορές

- Buchanan, B.G. (1983). Partial bibliography of work on expert systems. *Sigart Newsletter*, 84, 45-50.
- Coen, M. (1994). *SodaBot: A Software Agent Environment and Construction System* [Technical Report 1493]. Μασαχουσέτη: MIT AI Lab.
- Hayes-Roth, B. (1990). Architectural foundations for real-time performance in intelligent agents. *Real-Time Systems*, 2 (1-2), 99-125
- Hayes – Roth, B. (1995). An architecture for adaptive intelligent systems. *Artificial Intelligence (Special Issue on Agents and Interactivity)*, 72, 329-365.
- Huhns, M. & Stephens, L. (1999). Multiagent systems and societies of agents. Στο G.Weiss (εκδ.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (σ.79-122). Μασαχουσέτη, ΗΠΑ : MIT Press . Ανακτήθηκε από http://www.uma.ac.ir/files/site1/a_akbari_994c8e8/gerhard_weiss___multiagent_systems___a_modern_approach_to_distributed_artificial_intelligence.pdf

Maes, P.(1995). Artificial Life Meets Entertainment: Life like Autonomous Agents. *Communications of the ACM*, 38 (11), 108-114.

Poole, D. & Mackworth, A. (2010). *Artificial Intelligence: Foundation of Computational Agents*. Καίμπριτζ, Βρετανία: Cambridge University Press.

Διαθέσιμο στην ιστοσελίδα <http://artint.info/html/ArtInt.html>

Russel, S. & Norvig, P. (2009). *Artificial Intelligence: A modern approach* (3^η έκδοση). Upper Saddle River, Νιου Τζέρσεϊ: Pearson Education. Ανακτήθηκε από <http://51lica.com/wp-content/uploads/2012/05/Artificial-Intelligence-A-Modern-Approach-3rd-Edition.pdf>

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, 338-353.

Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Μασαχουσέτη, ΗΠΑ: MIT Press.

Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κοκκοράς, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3^η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστημίου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Ένας λογισμικός πράκτορας με εσωτερική κατάσταση διαθέτει υποχρεωτικά :

1. δυνατότητες μάθησης,
2. εσωτερική συμβολική αναπαράσταση του περιβάλλοντος,
3. κανόνες για το σχεδιασμό της επόμενης ενέργειας.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 2 και 3.

Κριτήριο αξιολόγησης 2

Ποια είναι τα χαρακτηριστικά των συνεργαζόμενων πρακτόρων σε ένα πολυπρακτορικό σύστημα;

1. Όλοι οι πράκτορες έχουν πλήρη πληροφορία.
2. Δεν υπάρχει κεντρικός έλεγχος στο σύστημα.
3. Τα δεδομένα είναι καταμεμημένα.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 2 και 3.

Κριτήριο αξιολόγησης 3

Ποια από τα παρακάτω αποτελούν κοινά χαρακτηριστικά νοημόνων πρακτόρων;

1. εκδήλωση πεποιθήσεων και επιθυμιών,
2. «ανθρωπόμορφη» αρχιτεκτονική λογισμικού,
3. χρήση έμπειρου συστήματος βασισμένου σε κανόνες,
4. ύπαρξη κάποιου βαθμού «νοημοσύνης» μέσα στο πρόγραμμα,

5. χρήση κινητών μελών.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 2 και 4.

Κριτήριο αξιολόγησης 4

Ποια από τα παρακάτω αποτελούν βασική υποδιαίρεση πρακτόρων;

1. έμπειροι,
2. αντιδραστικοί,
3. παραγωγικοί,
4. σταθεροί,
5. ορθολογιστικοί.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 2 και 5.

Κριτήριο αξιολόγησης 5

Ποιά από τα παρακάτω προδιορίζει ένα πολυπρακτορικό σύστημα;

1. Σύνολο από πράκτορες που δρουν αυτόνομα, για να επιλύσουν προβλήματα που είναι πέρα των δυνατοτήτων και της γνώσης ενός μόνο πράκτορα.
2. Όλοι οι συμμετέχοντες πράκτορες δρουν μαζί, για να επιλύσουν προβλήματα που είναι πέρα των δυνατοτήτων και της γνώσης ενός μόνο πράκτορα.
3. Ένας μόνο πράκτορας προσπαθεί να δράσει, ώστε να λύσει τα προβλήματα άλλων πρακτόρων που δεν έχουν την γνώση και την ικανότητα να το κάνουν μόνοι τους.

Απάντηση/Λύση

Σωστή απάντηση είναι η 2.

ΚΕΦΑΛΑΙΟ 7 - Επεξεργασία και Κατανόηση Φυσικής Γλώσσας

Σύνοψη

Η επεξεργασία φυσικής γλώσσας είναι μια από τις πιο δημοφιλείς εφαρμογές της TN. Η ιδέα του να επικοινωνεί κάποιος με τον υπολογιστή και να τον ελέγχει μιλώντας τη μητρική του γλώσσα ή κάποια ευρύτερα ομιλούμενη, όπως τα αγγλικά, είναι πολύ ελκυστική. Όμως, η φυσική γλώσσα έχει διττή φύση (ως προς τη σύνταξη και ως προς τη σημασιολογία), γεγονός που δεν εμποδίζει μεν την επεξεργασία της, αλλά δημιουργεί προβλήματα στην κατανόησή της, με αποτέλεσμα να καθίσταται το εγχείρημα της επεξεργασίας και παράλληλα της κατανόησής της ιδιαίτερα δύσκολο. Στο κεφάλαιο αυτό εξετάζουμε τα στάδια της επεξεργασίας φυσικής γλώσσας (: τη συντακτική, τη σημασιολογική και την πραγματολογική ανάλυση) και θίγουμε θέματα σχετικά με την πραγματολογική ανάλυση και τις τεχνικές που χρησιμοποιούνται για διευκρίνιση διφορούμενων προτάσεων και καθιστούν εφικτή την κατανόηση κειμένου.

Προαπαιτούμενη γνώση

Επεξεργασία γλωσσών προγραμματισμού, Μεταγλωττιστές

7.1 Εισαγωγή

Πριν κάνουμε επισκόπηση στο πώς μπορούμε να επιτύχουμε την **επεξεργασία φυσικής γλώσσας** (natural language processing-NLP) και την **κατανόηση φυσικής γλώσσας** (natural language understanding-NLU), θα παρουσιάσουμε τα οφέλη που μπορεί να επιφέρει η χρήση φυσικής γλώσσας κατά την επικοινωνία χρήστη-υπολογιστή. Πλήθος τομέων μπορούν να επωφεληθούν από τη χρήση της, με κυριότερο, καταρχάς, την **επικοινωνία ανθρώπου-μηχανής** (human-computer interaction). Στο χώρο αυτό, η χρήση φυσικής γλώσσας επιτρέπει στους χρήστες να χρησιμοποιούν το φυσικό τρόπο επικοινωνίας τους και όχι τεχνητές γλώσσες (προγραμματισμού, μηχανής, κ.ά.) ή δομημένα μενού. Μια τέτοια προσέγγιση έχει και προτερήματα και μειονεκτήματα. Ναι μεν δεν απαιτείται εκπαίδευση στη χρήση της γλώσσας, αλλά αυτό διευκολύνει περισσότερο τους περιστασιακούς χρήστες και λιγότερο τους εξειδικευμένους, όπως είναι για παράδειγμα οι προγραμματιστές ή οι υπάλληλοι γραφείου που εισάγουν στοιχεία σε φόρμες.

Μια δεύτερη περιοχή είναι αυτή της **διαχείρισης πληροφορίας** (information management), όπου η NLP θα μπορούσε να ενεργοποιήσει διαδικασίες αυτόματης διαχείρισης και επεξεργασίας της πληροφορίας με βάση τη διερμηνεία της. Αν, για παράδειγμα, ένα σύστημα μπορούσε να κατανοήσει το νόημα ενός εγγράφου, θα μπορούσε να το αρχειοθετήσει μαζί με τα άλλα αντίστοιχα έγγραφα.

Τρίτη περιοχή είναι αυτή της **αναζήτησης σε βάσεις δεδομένων** (database searching). Οι συνήθεις τρόποι έκφρασης μιας επιθυμητής πληροφορίας είναι μέσω επιλογής από λίστες, συμπλήρωσης μενού ή σύνταξης του αιτήματος σε τεχνητή γλώσσα (special query language-SQL). Η χρήση τεχνητής γλώσσας επιτρέπει μεν την ανάπτυξη απλών μηχανισμών αναζήτησης, αλλά και πάλι ο χρήστης πρέπει να έχει κάποια γνώση σχετικά με τη δομή της βάσης. Από την άλλη πλευρά, ο χρήστης είναι πιο εξοικειωμένος με το περιεχόμενο ή την περιοχή ενδιαφέροντος της βάσης παρά με τη δομή της. Με τη χρήση φυσικής γλώσσας, τα αιτήματα μπορεί να περιοριστούν σε όρους σχετικούς με το περιεχόμενο και την περιοχή ενδιαφέροντος.

7.2 Δυσκολίες στην Κατανόηση Φυσικής Γλώσσας

Η μεγαλύτερη δυσκολία στην επεξεργασία φυσικής γλώσσας είναι η διφορούμενη ερμηνεία που προκαλεί **ασάφεια στη γλώσσα** (ambiguity of language) σε πολλά επίπεδα: .

- καταρχάς, **ασάφεια σε επίπεδο σύνταξης** (ambiguity at syntactic level) της γλώσσας. Κάποιες συντακτικά ορθά προτάσεις επιδέχονται πάνω από μια διερμηνεία, ανάλογα με το πώς θα αναλυθούν συντακτικά, καθιστάμενες συντακτικά ασαφείς. Για παράδειγμα:

Χτύπησα τον κλέφτη με το τσεκούρι. Το τσεκούρι ήταν το όπλο με το οποίο χτύπησα τον κλέφτη ή χτύπησα τον κλέφτη που κρατούσε το τσεκούρι;

- Δευτερευόντως, **ασάφεια σε επίπεδο λεξιλογικό** (ambiguity at lexical level), όταν το νόημα μιας λέξης είναι διφορούμενο. Για παράδειγμα:

Το πρώτο γράμμα του Γιώργου. Εννοεί το πρώτο γράμμα που έγραψε ο Γιώργος ή το γράμμα του αλφαβήτου από το οποίο αρχίζει το όνομα «Γιώργος»; Η λέξη γράμμα έχει δυο έννοιες, της επιστολής και του γράμματος του αλφαβήτου.

- Τρίτον, **ασάφεια σε αναφορικό επίπεδο** (ambiguity at referential level), όταν δεν είναι ευκρινές το σε ποιον, πού ή σε τι η πρόταση αναφέρεται. Για παράδειγμα:

Ο Γιάννης χτύπησε τον Πέτρο, γιατί του αρέσει η Μαίρη. Σε ποιον αρέσει η Μαίρη, στο Γιάννη ή στον Πέτρο;

- Τέταρτον, **ασάφεια σε σημασιολογικό επίπεδο** (ambiguity at semantic level), όταν, με διατήρηση της ίδιας συντακτικής ανάλυσης, η πρόταση επιδέχεται τουλάχιστον δυο διαφορετικές ερμηνείες. Για παράδειγμα:

Τον άφησε στα κρύα του λουτρού. Η πρόταση κυριολεκτεί ότι κάποιος άφησε κάποιον άλλον στα κρύα ενός λουτρού ή παρουσιάζει μεταφορικά ότι τον παράτησε και έφυγε στη μέση κάποιας συνεργασίας;

- Τέλος, **ασάφεια σε πραγματολογικό επίπεδο** (pragmatic level), κατά τη διερμηνεία μιας πρότασης, όταν λαμβάνουμε υπόψη το πλαίσιο του κειμένου που την περιέχει. Στην παρακάτω πρόταση δεν είναι εύκολο να λυθεί η πραγματολογική ασάφεια:

Οι δεινόσαυροι έχουν εξαφανιστεί πολλά χρόνια. Πόσα χρόνια είναι τα πολλά χρόνια;

Οι επιμέρους ασάφειες μπορεί να συνδυαστούν σε μια μόνη πρόταση. Για παράδειγμα, η παρακάτω πρόταση περιέχει ασάφεια σε λεξιλογικό, αναφορικό και πραγματολογικό επίπεδο:

- *Ο Νίκος ζήτησε από τον Ηλία να τον αντικαταστήσει στη δουλειά σήμερα, πριν φύγει ταξίδι.* Υπάρχει αμφιβολία σχετικά με την απόδοση του χρονικού προσδιορισμού "σήμερα" σε ένα από τα δύο ρήματα, όπως και για το ποιος θα φύγει ταξίδι (πραγματολογικό και αναφορικό επίπεδο).
- *Ο Γιάννης χτύπησε εχθές τον Πέτρο, που είναι γάιδαρος, επειδή του αρέσει η Μαίρη.* Σε επίπεδο σημασιολογικής ασάφειας δεν ξέρουμε αν ο Πέτρος είναι ζώο γάιδαρος ή άνθρωπος που συμπεριφέρεται ως γάιδαρος. Από πλευράς αναφορικής δεν ξέρουμε σε ποιον αρέσει η Μαίρη και από πλευράς πραγματολογικής δεν ξέρουμε το πότε είναι το «χθες».

7.3 Ιστορική Αναδρομή

Από το 1950-1960, οπότε κατασκευάζονται οι πρώτοι υπολογιστές και επιτρέπουν το χειρισμό αριθμών και λέξεων, παρουσιάζονται και οι πρώτες δυνατότητες επεξεργασίας γλωσσικής γνώσης.

- Στα μέσα της δεκαετίας του 1960, στο σύστημά του ELIZA ο Weizenbaum βασίζεται στα αυτόματα πεπερασμένων καταστάσεων (*finite state automata, FSA*) για την αναγνώριση λεκτικών οντοτήτων (π.χ., λέξεων και άλλων στοιχείων του λόγου). Τη ίδια δεκαετία, ο

Schank με τη βοήθεια των *σημασιολογικών δικτύων (semantic networks)* εισάγει την έννοια των *εννοιολογικών εξαρτήσεων (conceptual dependencies)*.

- Τα έτη 1968-1970, ο Winograd, με το σύστημά του *SHRDLU*, γραμμένο σε Lisp, προσφέρει το πρώτο σύστημα κατανόησης φυσικής γλώσσας.
- Στις αρχές της δεκαετίας του 1970 ο Montague εργάστηκε επί της αρχής της συνθετικότητας, για να αναλύσει προτάσεις της φυσικής γλώσσας με αρκετή επιτυχία. Ήδη στα τέλη του 19^{ου} και στις αρχές του 20^{ου} αιώνα, ο Frege είχε εισαγάγει την ιδέα πως η σημασία μιας πρότασης είναι συνθετική και απαρτίζεται από τις *επιμέρους έννοιες των συνθετικών μερών* της (*compositional semantics*).
- Το 1977 οι Schank και Abelson συνέβαλαν σημαντικά στην κατανόηση της φυσικής γλώσσας με την αναπαράσταση του κειμένου μέσω *σεναρίων (scripts)* που εξετάζουν την αιτιότητα και τη διαδοχή των γεγονότων μέσα σε αυτό.

Μπορεί σήμερα η επεξεργασία και κατανόηση φυσικής γλώσσας να επιφέρει εντυπωσιακά αποτελέσματα, αλλά στις απαρχές του κλάδου οι τεχνικές που χρησιμοποιούνταν και τα αποτελέσματα ήταν στοιχειώδη. Για παράδειγμα, στο πρόγραμμα ELIZA, το οποίο προσομοιώνει τη συνομιλία ενός ψυχιάτρου με μια ασθενή, γίνεται απλή αναγνώριση λεκτικών μονάδων που πυροδοτούν τυποποιημένες απαντήσεις του προγράμματος-ψυχιάτρου. Ακολούθως, δίνεται ένας παραδειγματικός διάλογος:

ΟΙ ΑΝΘΡΩΠΟ ΕΙΝΑΙ ΟΙΛΟΙ ΙΔΙΟΙ : ασθενής
ΜΕ ΠΟΙΟ ΤΡΟΠΟ; : πρόγραμμα-ψυχίατρος
ΠΑΝΤΑ ΑΠΑΙΤΟΥΝ ΚΑΤΙ ΑΠΟ ΕΜΑΣ ΜΕ ΤΟΝ ΕΝΑ Η ΤΟΝ ΑΛΛΟ
ΤΡΟΠΟ
ΜΠΟΡΕΙΣ ΝΑ ΣΚΕΦΤΕΙΣ ΕΝΑ ΣΥΓΚΕΚΡΙΜΕΝΟ ΠΑΡΑΔΕΙΓΜΑ;
ΝΑ, Ο ΦΙΛΟΣ ΜΟΥ ΜΕ ΕΚΑΝΕ ΝΑ ΕΡΘΩ ΕΔΩ
Ο ΦΙΛΟΣ ΣΟΥ ΣΕ ΕΚΑΝΕ ΝΑ ΕΡΘΕΙΣ ΕΔΩ;
ΛΕΕΙ ΟΤΙ ΕΊΜΑΙ ΣΕ ΚΑΤΑΘΛΙΞΗ ΤΟΝ ΠΕΡΙΣΣΟΤΕΡΟ ΧΡΟΝΟ
ΛΥΠΑΜΑΙ ΠΟΥ ΑΚΟΥΩ ΟΤΙ ΕΙΣΑΙ ΣΕ ΚΑΤΑΘΛΙΨΗ
ΕΙΝΑΙ ΑΛΗΘΕΙΑ, ΕΙΜΑ ΔΥΣΤΥΧΗΣ
ΜΟΝΙΖΕΙΣ ΟΤΙ ΕΡΧΟΜΕΝΗ ΕΔΩ ΘΑ ΠΑΨΕΙΣ ΝΑ ΕΙΣΑΙ ΔΥΣΤΥΧΗΣ;
ΧΡΕΙΑΖΟΜΑΙ ΚΑΠΟΙΑ ΒΟΗΘΕΙΑ, ΑΥΤΟ ΕΙΝΑΙ ΣΙΓΟΥΡΟ
ΤΙ ΘΑ ΣΗΜΑΙΝΕ ΓΙΑ ΣΕΝΑ ΝΑ ΠΑΡΕΙΣ ΚΑΠΟΙΑ ΒΟΗΘΕΙΑ;
ΙΣΩΣ ΘΑ ΜΕ ΒΟΗΘΟΥΣΕ ΝΑ ΤΑ ΒΓΑΛΩ ΠΕΡΑ ΜΕ ΤΗ ΜΗΤΕΡΑ ΜΟΥ
ΠΕΣ ΜΟΥ ΠΕΡΙΣΣΟΤΕΡΑ ΓΙΑ ΤΗΝ ΟΙΚΟΓΕΝΕΙΑ ΣΟΥ
Η ΜΗΤΕΡΑ ΜΟΥ ΜΕ ΦΡΟΝΤΙΖΕΙ
ΠΟΙΟΣ ΑΛΛΟΣ ΣΤΗΝ ΟΙΚΟΓΕΝΕΙΑ ΣΟΥ ΣΕ ΦΡΟΝΤΙΖΕΙ;
Ο ΠΕΑΤΕΡΑΣ ΜΟΥ
Ο ΠΑΤΕΡΑΣ ΣΟΥ;
ΜΟΙΑΖΕΙΣ ΜΕ ΤΟΝ ΠΑΤΕΡΑ ΜΟΥ ΚΑΤΑ ΚΑΠΟΙΟΝ ΤΡΟΠΟ
ΠΟΙΑ ΟΜΟΙΟΤΗΤΑ ΒΛΕΠΕΙΣ;
ΔΕΝ ΕΙΣΑΙ ΕΠΙΘΕΤΙΚΟΣ, ΑΛΛΑ ΝΟΜΙΖΩ ΟΤΙ ΔΕ ΘΕΛΕΙΣ ΝΑ ΣΤΟ
ΣΗΜΕΙΩΣΩ
ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΔΕΝ ΕΙΜΑΙ ΠΟΛΥ
ΕΠΙΘΕΤΙΚΟΣ;

ΔΕ ΔΙΑΦΩΝΕΙΣ ΜΑΖΙ ΜΟΥ
ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΔΕ ΔΙΑΦΩΝΩ ΜΑΖΙ ΣΟΥ;
ΜΕ ΦΟΒΑΣΑΙ
ΣΕ ΕΥΧΑΡΙΣΤΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΣΕ ΦΟΒΑΜΑΙ;
Ο ΠΑΤΕΡΑΣ ΜΟΥ ΤΟΥΣ ΦΟΒΑΤΑΙ ΟΛΟΥΣ
ΤΙ ΑΛΛΟ ΣΟΥ ΕΡΧΕΤΑΙ ΣΤΟ ΜΥΑΛΟ ΟΤΑΝ ΣΚΕΦΤΕΣΑΙ ΤΟΝ
ΠΑΤΕΡΑ ΣΟΥ;
ΨΕΥΡΟΠΑΛΛΗΚΑΡΙΣΜΟΙ
ΕΧΕΙ ΑΥΤΟ ΚΑΠΟΙΑ ΣΧΕΣΗ ΜΕ ΤΟ ΓΕΓΟΝΟΣ ΟΤΙ Ο ΦΙΛΟΣ ΣΟΥ
ΣΕ ΕΚΑΝΕ ΝΑ ΕΡΘΕΙΣ ΕΔΩ;

Οι τεχνικές του ELIZA είναι οι ακόλουθες:

A. Οι απαντήσεις συνθέτονται μέσω περιγραμμάτων (templates) ή σκελετών, π.χ.

Περίγραμμα 1:

ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ XXX ?

Περίγραμμα 2:

ΠΕΣ ΜΟΥ ΠΕΡΙΣΣΟΤΕΡΑ ΓΙΑ XXX ?

Περίγραμμα 3:

ΜΕ ΣΕ ΕΥΧΑΡΙΣΤΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ XXX ?

B. Η πρόταση εισόδου εξετάζεται για την παρουσία λέξεων-κλειδιών ή μερών προτάσεων που αναγνωρίζονται από αυτόματα πεπερασμένων καταστάσεων, π.χ.:

ΕΓΩ, ΕΣΥ, ΜΗΤΕΡΑ, ΜΙΣΩ, ...

Γ. Κάθε λέξη-κλειδί σχετίζεται με ένα σύνολο κανόνων που μετασχηματίζουν τη φράση εισόδου σε φράση εξόδου, π.χ.:

Κανόνας 1:

ΕΣΥ ΔΕΝ ΖΖΖ ΜΕ ΕΜΕΝΑ → XXX ΕΓΩ ΔΕΝ ΖΖΖ ΜΕ ΕΣΕΝΑ;

Κανόνας 2:

ΕΙΣΑΙ ΖΖΖ → ΕΙΜΑΙ ΖΖΖ;

Όπου XXX: ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ή
ΣΕ ΕΥΧΑΡΙΣΤΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ

Παράδειγμα εφαρμογής των κανόνων:

ΔΕ ΔΙΑΦΩΝΕΙΣ ΜΑΖΙ ΜΟΥ

ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΔΕ ΔΙΑΦΩΝΩ ΜΑΖΙ ΣΟΥ;

ΜΕ ΦΟΒΑΣΑΙ

ΣΕ ΕΥΧΑΡΙΣΤΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΣΕ ΦΟΒΑΜΑΙ;

Όταν το σύστημα δεν βρίσκει λέξεις-κλειδιά, τότε αντιδρά με φράση από προηγούμενη συζήτηση.

Όταν, επίσης, βλέπει να δίνονται μονολεκτικές απαντήσεις, τότε αντιδρά με κάποια γενικής φύσης φράση, όπως είναι:

ΔΕ ΦΑΙΝΕΣΑΙ ΠΟΛΥ ΟΜΙΛΗΤΙΚΟΣ ΣΗΜΕΡΑ

7.4 Πώς πραγματοποιείται η Επεξεργασία Φυσικής Γλώσσας σήμερα

Υπάρχουν τρία στάδια για την επεξεργασία φυσικής γλώσσας: συντακτική ανάλυση, σημασιολογική ανάλυση και η πραγματολογική ανάλυση. Οι προτάσεις μπορεί να είναι **καλοσηματισμένες** (well formed) ή **κακοσηματισμένες** (ill formed) σε επίπεδο σύνταξης, σημασιολογίας ή πραγματολογικής διαφάνειας. Για παράδειγμα, ας δούμε περιπτώσεις καλοσηματισμένων και κακοσηματισμένων απαντήσεων στο ερώτημα : ‘Ξέρετε πού είναι το σχολείο;’

- *Το σχολείο είναι στο βάθος του δρόμου:* καλοσηματισμένη πρόταση από όλες τις απόψεις και θα μπορούσε να δοθεί κάλλιστα ως απάντηση στο ερώτημα.
- *Το σχολείο είναι στο βάθος του ελέφαντα:* συντακτικά καλοσηματισμένη πρόταση, αλλά σημασιολογικά κακοσηματισμένη. Ξέρουμε από τις γνώσεις μας σε σχέση με το τι σημαίνει σχολείο και τι ελέφαντας ότι από την πρόταση δεν βγαίνει νόημα.
- *Το σχολείο βάθος δρόμου είναι.* Αποτελεί μια συντακτικά κακοσηματισμένη πρόταση.
- *Ναι.* Αποτελεί μια πραγματολογικά κακοσηματισμένη πρόταση. Δεν απαντά στο ερώτημα.

Κατά τη διάρκεια της επεξεργασίας το σύστημα θα προσδιορίσει κατά πόσο η πρόταση είναι καλοσηματισμένη. Τα τρία στάδια επεξεργασίας δεν είναι κατ’ ανάγκη διακριτά ή σειριακά, αλλά θα τα θεωρήσουμε ως τέτοια για τις ανάγκες μιας όσο γίνεται πιο κατανοητής παρουσίασης.

Για να μπορέσουν να λειτουργήσουν τα στάδια της επεξεργασίας, είναι απαραίτητη η ύπαρξη ενός λεξικού της φυσικής γλώσσας, όπου θα είναι αποτυπωμένη η έννοια κάθε λέξης και επιπλέον στοιχεία που είναι απαραίτητα κατά την ανάλυση.

7.4.1 Λεξικό

Το λεξικό μιας φυσικής γλώσσας περιέχει όλες τις λέξεις που χρησιμοποιούνται στη γλώσσα αυτή. Λόγω του μεγάλου αριθμού των λέξεων, αποθηκεύεται μόνο η βασική μορφή κάθε λέξης και οι άλλες μορφές προκύπτουν με κανόνες **μορφολογικής ανάλυσης** (morphological analysis):

- πρόσωπα, χρόνοι, πτώσεις, γένη, αριθμοί (**μορφολογία κλίσεων** - inflectional morphology).
- νέες λέξεις, με προσθήκη γνωστών προθεμάτων (π.χ. στερητικό α-) ή καταλήξεων (**ετυμολογική μορφολογία** - derivational morphology).
- σύνθετες λέξεις (**σύνθεση λέξεων** - compounding).

Επιπλέον, το λεξικό πρέπει να κατατάσσει **σημασιολογικά** όλες τις λέξεις ανάλογα με τον τρόπο χρήσης τους. Για παράδειγμα:

- Το κάθε ρήμα θα πρέπει να σχετίζεται με την κατηγορία στην οποία ανήκει το
 - ο υποκείμενό του και το
 - ο αντικείμενό του (όταν το ρήμα είναι μεταβατικό)

Η γνώση μιας τέτοιας πληροφορίας δε θα επιτρέψει να παραχθούν φράσεις όπως «*Ο φοιτητής είναι Μαρία*».

Τα λεξικά που οργανώνονται σημασιολογικά ονομάζονται **οντολογίες** (ontologies).

7.5 Συντακτική Ανάλυση

Η **συντακτική ανάλυση** (syntactic analysis) εξετάζει τη δομή των προτάσεων προσπαθώντας να διαπιστώσει με διάφορους τρόπους αν αυτή είναι ορθή ή λανθασμένη.

Η πιο απλή προσέγγιση είναι η χρήση κάποιου τύπου ταιριάσματος προτύπου, οπότε απαιτείται να υπάρχουν αποθηκευμένα πρότυπα προτάσεων που περιέχουν μεταβλητές, ώστε να επιτρέπεται το ταιρίασμα με συγκεκριμένες προτάσεις, Π.χ. το πρότυπο

<η ** πίνει **>

όπου ** ταιριάζει με οτιδήποτε, επιτρέπει την αναγνώριση ως συντακτικά καλοσχηματισμένων προτάσεων όπως: *ο Γιώργος οδηγεί ένα ωραίο αυτοκίνητο, ο Πέτρος οδηγεί το τραίνο*, κλπ. Παρά ταύτα, η προσέγγιση δε θεωρείται πολύ αποτελεσματική, γιατί δεν επιτρέπει την αναγνώριση προβληματικών σημασιολογικά προτάσεων, όπως, π.χ., *ο τοίχος οδηγεί ένα σπίτι*. Η μέθοδος του ταιριάσματος προτύπων θεωρείται κατάλληλη μόνο για πολύ στοιχειώδη περιβάλλοντα επικοινωνίας, όπου η χρήση της φυσικής γλώσσας πραγματοποιείται υπο περιορισμούς. Χρησιμοποιήθηκε αρχικά από το πρόγραμμα ELIZA (Weizenbaum, 1966) που, όπως αναφέραμε και στην [προηγούμενη παράγραφο](#), δεν κατόρθωσε να εντοπίσει τις διφορούμενες προτάσεις και κατέληξε να αποδέχεται ως καλοσχηματισμένες προτάσεις ασυνάρτητες.

Πιο αποτελεσματική μέθοδος είναι η **γραμματική ανάλυση** μιας πρότασης (parsing), κατά την οποία η πρόταση μετατρέπεται σε μια ιεραρχική δομή που υποδεικνύει τα συστατικά της μέρη. Τα συστήματα γραμματικής ανάλυσης αποτελούνται από:

- μια **γραμματική** που προσφέρει μια δηλωτική αναπαράσταση των συντακτικών στοιχείων σχετικά με τη γλώσσα,
- έναν **συντακτικό αναλυτή** που συγκρίνει την πρόταση που εισάγεται με τη γραμματική.

Οι συντακτικοί αναλυτές μπορεί να είναι ανοδικοί, εάν αρχίζουν από τα σύμβολα εισόδου και προσπαθούν να ταιριάξουν κανόνες σε αυτά, ή καθοδικοί, οπότε αρχίζουν από τους γραμματικούς κανόνες και προσπαθούν να ταιριάξουν τα σύμβολα εισόδου σε αυτούς.

Άλλες μέθοδοι συντακτικής ανάλυσης περιλαμβάνουν **διαγράμματα μετάβασης** (transition networks) και **επαυξημένα διαγράμματα μετάβασης** (augmented transition networks) (ενδεικτικά).

Αν υπάρχει αμφισημία στις λέξεις, συνήθως δοκιμάζεται η συντακτική ανάλυση διάφορων ερμηνειών των λέξεων μέσα από το λεξικό και επιλέγονται εκείνες που ταιριάζουν καλύτερα τόσο στη συντακτική όσο και στη σημασιολογική και πραγματολογική ανάλυση μιας πρότασης.

7.6 Γραμματική

Μια **γραμματική** (grammar) προσδιορίζει τις αποδεκτές δομές μιας γλώσσας. Αποτελείται από ένα σύνολο κανόνων του τύπου

A → B

που επιτρέπουν κάθε στοιχείο που ταιριάζει με το αριστερό μέρος τους να αντικαθίσταται από το δεξί μέρος. Για παράδειγμα, έστω ότι έχουμε δυο κανόνες:

1^{ος} κανόνας: υποκείμενο → άρθρο ουσιαστικό

2^{ος} κανόνας: πρόταση → υποκείμενο ρήμα αντικείμενο

τότε, στον 2^ο κανόνα, εκεί όπου παρουσιάζεται το αριστερό μέρος του 1^{ου} κανόνα, δηλαδή ο όρος 'υποκείμενο', μπορεί να αντικατασταθεί από το δεξί μέρος του, διαμορφώνοντας τον 2^ο κανόνα ως εξής:

πρόταση → άρθρο ουσιαστικό ρήμα αντικείμενο

Μια γραμματική έχει τρία συστατικά μέρη:

- **τερματικά σύμβολα,**
- **μη-τερματικά σύμβολα,**
- **κανόνες παραγωγής.**

Τερματικά σύμβολα (terminals) είναι οι λέξεις που αποτελούν τη γλώσσα και διαμορφώνουν το **λεξικό** της (lexicon). Επιπλέον, για κάθε λέξη στο λεξικό προσδιορίζεται ο θεμελιώδης τύπος της.

Παράδειγμα, η πρόταση η γάτα πίνει γάλα, περιέχει 4 τερματικά σύμβολα ('η', 'γάτα', 'πίνει', 'γάλα'), τα οποία στο λεξικό θα έχουν καταχωρηθεί με τους προσδιορισμούς τους ως εξής:

[η]: άρθρο

[γάτα]: ουσιαστικό

[πίνει]: ρήμα

[γάλα]: αντικείμενο

Τα **μη-τερματικά σύμβολα** (non-terminals) είναι ειδικά σύμβολα που περιγράφουν δομές της γλώσσας και είναι τριών τύπων:

- *λεκτικές κατηγορίες*, που είναι γραμματικές κατηγορίες λέξεων (π.χ. ουσιαστικό, ρήμα),
- *συντακτικές κατηγορίες*, που είναι επιτρεπόμενοι συνδυασμοί λεκτικών κατηγοριών (π.χ. αντικειμενική-πρόταση, κατηγορηματική-πρόταση, υποκείμενο),
- *αρχικό σύμβολο*, ένα ειδικό σύμβολο που αντιπροσωπεύει την αρχική πρόταση που περιγράφεται γραμματικά.

Οι **γραμματικοί κανόνες παραγωγής** ή **συντακτικοί κανόνες** (production rules or syntax rules) ή, απλώς, κανόνες περιγράφουν τους έγκυρους συνδυασμούς των λέξεων της γλώσσας. Ένας κανόνας έχει συνήθως την εξής μορφή:

$$A \rightarrow O\Phi \ P\Phi$$

όπου, A το αρχικό σύμβολο για μια ονοματική φράση OΦ (π.χ. η γάτα) και PΦ για μια ρηματική φράση (π.χ. πίνει γάλα)

Αν μια σωστή πρόταση δεν αναγνωρίζεται συντακτικά από τη γραμματική, τότε η γραμματική πρέπει να επεκτείνεται, ώστε να αναγνωρίζεται και η συγκεκριμένη πρόταση.

7.6.1 Γραμματικές Οριστικών Προτάσεων

Γραμματικές σαν την παραπάνω ονομάζονται **Γραμματικές Οριστικών Προτάσεων-ΓΟΠ** (Definite Clause Grammars-DCGs).

Τρία είναι τα κύρια προβλήματα που πρέπει να αντιμετωπιστούν, όταν γίνεται η χρήση τους από ένα σύστημα συντακτικής ανάλυσης:

- Η γραμματική μια φυσικής γλώσσας έχει πολλούς κανόνες και, άρα, χρειάζεται μεγάλη προσπάθεια για την αποτύπωσή τους.
- Κατά συνέπεια, η αναζήτηση της δομής μιας πρότασης καθίσταται πολύπλοκο πρόβλημα αναζήτησης.
- Είναι απαραίτητη η βοήθεια από σημασιολογική και πραγματολογική ανάλυση, για να εντοπίζονται σημεία μη εγκυρότητας μέσα σε συντακτικά ορθές προτάσεις.

Παράδειγμα γραμματικής με αποτύπωση μόνο συντακτικών κανόνων:

Γραμματική Γ1

πρόταση → υποκείμενο ρήμα (αντικείμενο | κατηγορούμενο)

υποκείμενο → άρθρο ουσιαστικό

αντικείμενο → άρθρο ουσιαστικό | ουσιαστικό

κατηγορούμενο → επίθετο | ουσιαστικό

Η γραμματική συνοδεύεται από το παρακάτω λεξικό:

[είναι]: ρήμα

[πίνει]: ρήμα

[γάτα]: ουσιαστικό

[έξυπνος]: επίθετο

[Γιάννης]: ουσιαστικό

[μαύρη]: επίθετο

[νέος]: επίθετο

[ο]: άρθρο

[η]: άρθρο

Στην παραπάνω γραμματική Γ1, το αρχικό σύμβολο είναι το πρόταση, τερματικά σύμβολα είναι όλα τα σύμβολα μεταξύ '[' και ']' που εμφανίζονται στο λεξικό, όλα δε τα υπόλοιπα είναι μη-τερματικά σύμβολα.

Βασικό πρόβλημα της παραπάνω γραμματικής είναι το ότι δεν μπορεί να αναγνωρίσει σημασιολογικά λάθη, όπως αυτά της πρότασης η Γιάννης πίνει γάτα. Η επίλυση του προβλήματος απαιτεί τον εμπλουτισμό της γραμματικής με επιπλέον κανόνες σημασιολογικής και πραγματολογικής μορφής. Άλλα πιθανά προβλήματα μπορεί να οφείλονται σε μη αναγνώριση λέξεων που είτε δεν υπάρχουν στο λεξικό είτε έχουν καταχωρηθεί λανθασμένα.

7.6.2 Συντακτικό Δένδρο

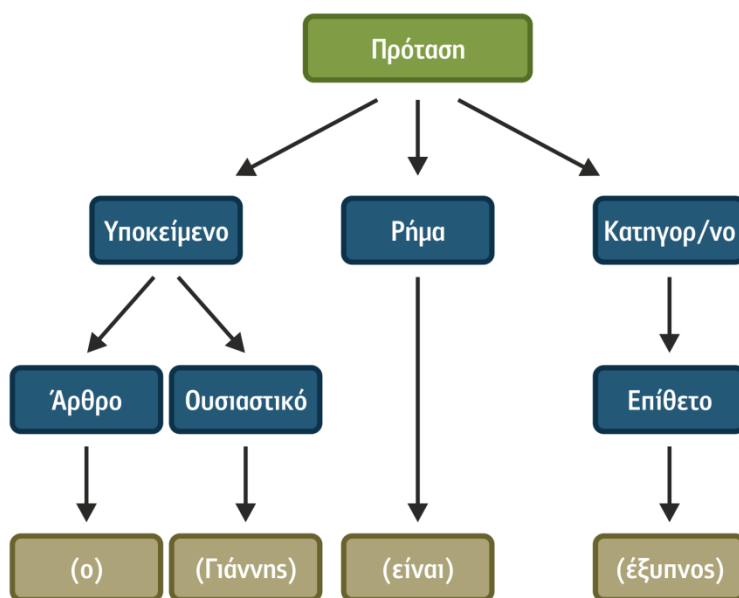
Η παραπάνω γραμματική Γ1 μπορεί να αναγνωρίσει συντακτικά οποιαδήποτε πρόταση είναι γραμμένη στη γλώσσα της. Ένας **συντακτικός αναλυτής** (parser) που κάνει καθοδική γραμματική ανάλυση (parsing) θα αρχίσει από το αρχικό σύμβολο της γραμματικής, στην περίπτωσή μας το πρόταση, το οποίο θα αποτελέσει τη ρίζα ενός δένδρου που καλείται **συντακτικό δένδρο** (parse tree). Ουσιαστικά, ένα συντακτικό δένδρο αντιπροσωπεύει την ιεραρχική διάσπαση μιας πρότασης στα συντακτικά μέρη της. Η ρίζα του είναι το αρχικό σύμβολο και κάθε ενδιάμεσος κόμβος του αντιπροσωπεύει ένα μη-τερματικό σύμβολο της γραμματικής. Τα φύλλα του δένδρου αντιπροσωπεύουν τερματικά σύμβολα.

Ο συντακτικός αναλυτής αναπτύσσει το συντακτικό δένδρο βήμα-βήμα με τη βοήθεια των παραγωγών της γραμματικής, έως ότου καταλήξει σε φύλλα, δηλαδή σε τερματικά σύμβολα. Σε κάθε βήμα ανάπτυξης, επιλέγεται ένας κόμβος που εκπροσωπεί ένα αριστερό μη-τερματικό σύμβολο ενός κανόνα και από κάτω διασπάται σε κόμβους που αντιπροσωπεύουν τα σύμβολα του δεξιού μέρους του κανόνα.

Το συντακτικό δένδρο που παράγει η συντακτική ανάλυση διερμηνεύεται

- από τη σημασιολογική ανάλυση ανάλογα με τα πιθανά νοήματα των δομικών της συστατικών,
- κατά την πραγματολογική ανάλυση, σύμφωνα με το περιεχόμενο (context) στο οποίο είναι ενταγμένη και τις προθέσεις που εκφράζονται μέσα σε αυτό.

Στο σχήμα 7.1 παρουσιάζεται το συντακτικό δένδρο που προκύπτει από την γραμματική ανάλυση της πρότασης ο Γιάννης είναι έξυπνος.



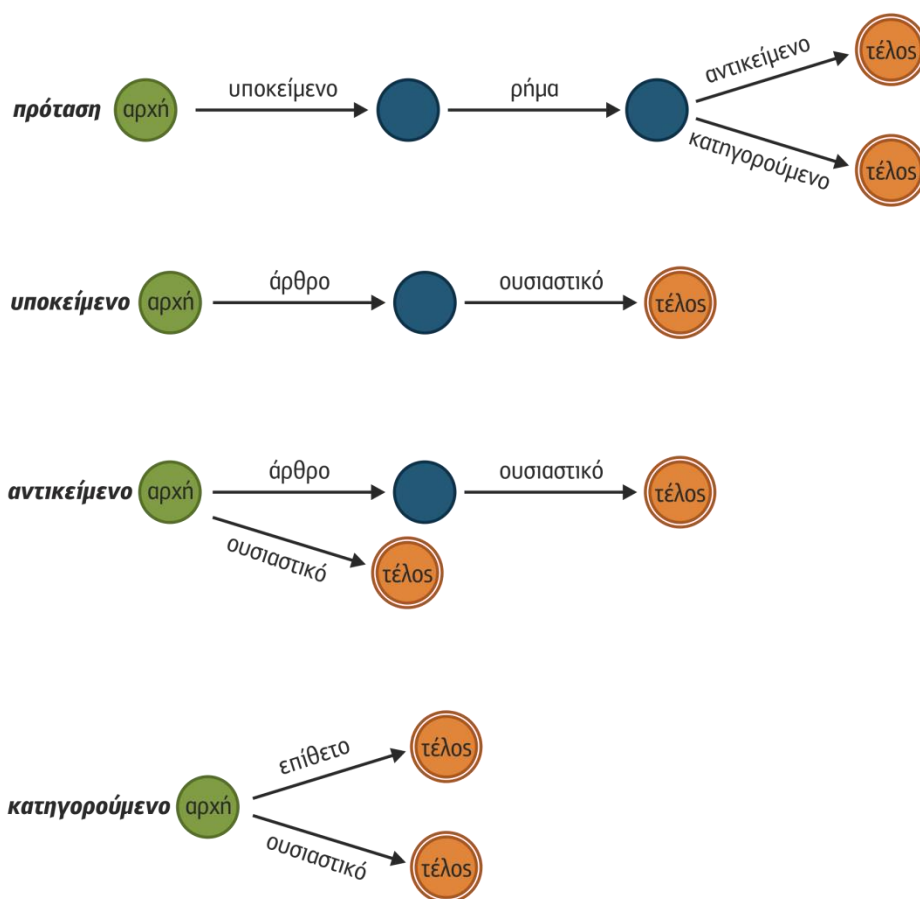
Σχήμα 7.1 Παράδειγμα συντακτικού δένδρου

Εντούτοις, η γραμματική που χρησιμοποιήθηκε είναι πολύ περιορισμένη. Για να την επεκτείνουμε, χρειαζόμαστε πολλούς ακόμα κανόνες και ένα πολύ μεγαλύτερο λεξικό. Θα επανέλθουμε στο θέμα αυτό αργότερα.

7.6.3 Διαγράμματα Μετάβασης

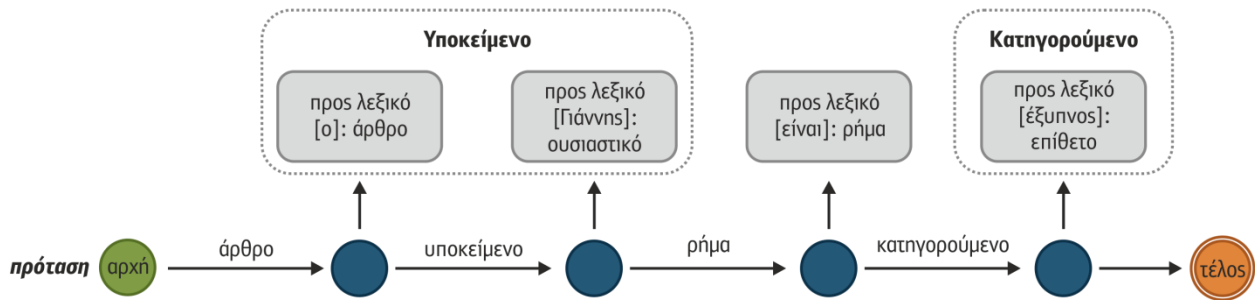
Ένα **διάγραμμα μετάβασης** (transition network) είναι μια μέθοδος συντακτικής ανάλυσης που αναπαριστά τη γραμματική ως ένα σύνολο μηχανών πεπερασμένων καταστάσεων. Μια μηχανή πεπερασμένων καταστάσεων, που καλείται επίσης **αυτόματο πεπερασμένων καταστάσεων** (finite state automaton-FSA), είναι ένα μοντέλο υπολογιστικής συμπεριφοράς, στο οποίο κάθε κόμβος αναπαριστά μια εσωτερική κατάσταση του συστήματος και οι ακμές παρουσιάζουν τον τρόπο μετάβασης από κατάσταση σε κατάσταση. Στην περίπτωση της επεξεργασίας φυσικής γλώσσας, οι ακμές αντιπροσωπεύουν είτε ένα τερματικό είτε ένα μη τερματικό σύμβολο. Οι γραμματικοί κανόνες αντιστοιχούν σε μια διαδρομή μέσα στο διάγραμμα. Για κάθε μη-τερματικό σύμβολο, υπάρχει και ένα διάγραμμα μετάβασης που το αναγνωρίζει.

Στο σχήμα 7.2 δίνεται ένα παράδειγμα χρήσης διαγραμμάτων μετάβασης για τη γραμματική Γ1.



Σχήμα 7.2 Παράδειγμα διαγραμμάτων μετάβασης

Η αναγνώριση της πρότασης ο Γιάννης είναι έξυπνος θα γίνει ως ακολούθως: Καταρχάς, θα επιχειρηθεί η διάσχιση του διαγράμματος του αρχικού συμβόλου πρόταση. Για τη διάσχιση του διαγράμματος απαιτείται, πρώτον, η αναζήτηση ενός τερματικού συμβόλου τύπου άρθρου. Η λέξη 'ο' είναι ορισμένη στο λεξικό ως άρθρο, οπότε επιτρέπει τη διάσχιση προς τον επόμενο κόμβο που απαιτεί ένα τερματικό σύμβολο τύπου ουσιαστικού. Η λέξη Γιάννης είναι ορισμένη στο λεξικό ως ουσιαστικό, οπότε η διάσχιση του διαγράμματος για την αναγνώριση του υποκειμένου ολοκληρώνεται επιτυχώς και επιστρέφουμε στο αρχικό διάγραμμα για την συνέχιση της διάσχισης. Ακολουθεί η ανάγκη διάσχισης του διαγράμματος του μη-τερματικού συμβόλου ρήμα. Για τη διάσχιση θα απαιτηθεί η αναζήτηση ενός τερματικού συμβόλου τύπου ρήματος. Η λέξη είναι που ακολουθεί είναι ορισμένη στο λεξικό ως ρήμα οπότε η διάσχιση είναι επιτυχής και προχωρούμε στην ανάγκη διάσχισης ή του διαγράμματος αντικείμενο ή του διαγράμματος κατηγορούμενο. Μετά από ανάλυση επιλέγεται το κατηγορούμενο, οπότε διασχίζοντας το διάγραμμά του απαιτείται η παρουσία ενός επιθέτου ή ενός ουσιαστικού. Δεδομένου ότι στην πρόταση ακολουθεί η λέξη έξυπνος που είναι καταχωρημένη ως επίθετο, η αναγνώριση ολοκληρώνεται επιτυχώς. Η ολοκληρωμένη συντακτική ανάλυση για την αναγνώριση της πρότασης ο Γιάννης είναι έξυπνος δίνεται στο σχήμα 7.3.



Σχήμα 7.3 Διάσχιση διαγραμμάτων μετάβασης κατά τη συντακτική ανάλυση

7.7 Αλγόριθμοι Διόρθωσης Ορθογραφικών Λαθών

Τα πιο κοινά λάθη που παρουσιάζει μια πρόταση και εμποδίζουν την αναγνώρισή της είναι τα ορθογραφικά. Για να αντιμετωπιστούν, η επεξεργασία φυσικής γλώσσας χρησιμοποιεί αλγόριθμους διόρθωσης λαθών που συνήθως έχουν τις παρακάτω δυνατότητες:

- κατανόηση γραπτού κειμένου,
- αναγνώριση λέξεων σε προφορικό λόγο,
- προγράμματα ορθογραφίας.

Η λειτουργία των αλγορίθμων διόρθωσης λαθών στηρίζεται στην έννοια της **εγγύτητας** (closeness) μεταξύ λέξεων. Τρόποι ορισμού της εγγύτητας είναι οι παρακάτω:

- μοντέλο χαρακτήρων,
- ηχητικό μοντέλο.

7.7.1 Μοντέλο Χαρακτήρων Διόρθωσης Λαθών

Στο **μοντέλο χαρακτήρων** (letter-based model), ως λάθη θεωρούνται:

- η εισαγωγή / διαγραφή ενός μεμονωμένου χαρακτήρα,
- η αντιμετάθεση δύο γειτονικών χαρακτήρων,
- η αντικατάσταση ενός γράμματος από ένα άλλο.

Αν θεωρήσουμε ότι η γλώσσα μας διαθέτει ένα αλφάβητο 24 γραμμάτων και ότι έχει συμβεί μόνο ένα λάθος σε μια λέξη, μια λέξη με 8 γράμματα μπορεί να έχει σε επίπεδο χαρακτήρων 9×24 πιθανά λάθη εισαγωγής, 8 πιθανά λάθη διαγραφής, 8×23 πιθανά λάθη αντικατάστασης και 7 πιθανά λάθη αντιμετάθεσης: άρα, μπορεί να έχει συνολικά 415 διαφορετικά λάθη.

Η χρήση της εγγύτητας χαρακτήρων μπορεί να βοηθήσει στον περιορισμό της αναζήτησης του λάθους. Για παράδειγμα, έστω ότι έχει γραφεί λανθασμένα η έγκυρη λέξη «αυτοκίνητο». Γειτονικές λανθασμένες λέξεις αυτής της λέξης, σε σχέση με την εγγύτητα είναι οι ακόλουθες:

με απόσταση 1 λάθος: «ατοκίνητο», «υατοκίνητο», «ααυτοκίνητο» κλπ,
 με απόσταση 2 λάθη: «αυτκοίνητο», «ατκίνητο», κλπ.

Όμως η διόρθωση λαθών βάσει εγγύτητας δε μπορεί να εφαρμοστεί σε όλες τις περιπτώσεις. Για παράδειγμα:

- Η λέξη «ατμοκίνητο» παρόλο που απέχει 2 λάθη από την έγκυρη «αυτοκίνητο» γίνεται εξαρχής αποδεκτή γιατί τα λάθη δε μπορούν να ανιχνευτούν εφόσον η λέξη «ατμοκίνητο» είναι επίσης έγκυρη !!!
- Η λανθασμένη λέξη «ατκίνητο» ισαπέχει από δύο έγκυρες λέξεις, τις «αυτοκίνητο» και «ατμοκίνητο» γεγονός που δημιουργεί πρόβλημα στη διόρθωση του λάθους.

7.7.2 Ηχητικό Μοντέλο Διόρθωσης Λαθών

Στο **ηχητικό μοντέλο** (sound-based model), οι λέξεις μεταφράζονται στο φωνητικό ισοδύναμο, στο οποίο διατηρείται όλη η πληροφορία που είναι απαραίτητη για την εκφώνηση της λέξης, χωρίς, ωστόσο, να διατηρείται και η ορθογραφία τους. Για παράδειγμα, γίνεται αντικατάσταση των «αι» από «ε», των «η» και «υ», από το «ι».

Αφού κατασκευαστεί το φωνητικό ισοδύναμο, μπορούν να εφαρμοστούν οι ίδιοι κανόνες αναζήτησης λαθών, όπως και στο μοντέλο χαρακτήρων. Όσο λιγότεροι είναι οι φθόγγοι, τόσο ευκολότερα μπορούν να ανιχνευθούν 'κοντινές' λέξεις.

7.8 Σημασιολογική ανάλυση

Κατά τη **σημασιολογική ανάλυση** (semantic analysis) πραγματοποιείται μετατροπή των προτάσεων σε εσωτερικές δομές αναπαράστασης γνώσης, χρησιμοποιώντας τη νοηματική σημασία των λέξεων. Για να επιτευχθεί κάτι τέτοιο, απαιτούνται οι λεγόμενες εξελιγμένες Γραμματικές Οριστικών Προτάσεων.

Σημαντικότερο πρόβλημα αποτελεί η ασάφεια σε επίπεδο λεκτικό που προκύπτει από την **πολυσημία** (ambiguity) των λέξεων. Ένας τρόπος αντιμετώπισης της ασάφειας αυτής είναι το να λαμβάνονται υπόψη χαρακτηριστικά της κλίσης των ουσιαστικών και ρημάτων: π.χ. *γένος, αριθμός, πτώση των ουσιαστικών, πρόσωπα ρημάτων*. Ακολουθεί ένα παράδειγμα εξελιγμένης γραμματικής που υποστηρίζει την παραπάνω προσέγγιση:

πρόταση → υποκειμένο (**Γένος, Αριθμός, ονομαστική**) ρήμα (**Αριθμός**)
αντικείμενο (**Γένος, Αριθμός, αιτιατική**)

πρόταση → υποκειμένο (**Γένος, Αριθμός, ονομαστική**) ρήμα (**Αριθμός**)
κατηγορούμενο (**Γένος, Αριθμός, ονομαστική**)

υποκειμένο → άρθρο (**Γένος, Αριθμός, Πτώση**)
ουσιαστικό (**Γένος, Αριθμός, Πτώση**)

κατηγορούμενο → επίθερο (**Γένος, Αριθμός, Πτώση**)

Σε γραμματικές του παραπάνω τύπου, τα χαρακτηριστικά διαδίδονται από τους ειδικότερους στους γενικότερους κανόνες, ώστε να υπάρχει συμφωνία των χαρακτηριστικών αυτών μεταξύ διαφορετικών τμημάτων της πρότασης. Για παράδειγμα:

- Ο αριθμός του υποκειμένου πρέπει να συμφωνεί με την κατάληξη του ρήματος.
- Το γένος, η πτώση και ο αριθμός του άρθρου του υποκειμένου πρέπει να συμφωνεί με τα αντίστοιχα χαρακτηριστικά της κλίσης του ουσιαστικού του υποκειμένου.

7.8.1 Αντιμετώπιση αμφιβολίας στην ερμηνεία

Η αμφιβολία στην ερμηνεία προκύπτει από την ασάφεια και σε σημασιολογικό και σε πραγματολογικό επίπεδο, όπως αναλύσαμε στην [εισαγωγική παράγραφο](#) του κεφαλαίου αυτού. Για

παράδειγμα, στην παρακάτω πρόταση υπάρχει αμφιβολία σχετικά με την απόδοση του χρονικού προσδιορισμού «σήμερα» σε ένα από τα δύο ρήματα:

Ο Νίκος ζήτησε από τον Ηλία να τον αντικαταστήσει στη δουλειά σήμερα.

Τέτοιας μορφής ασάφεια μπορεί να αντιμετωπισθεί με συντακτικούς κανόνες, όπως, για παράδειγμα, ότι χρονικοί προσδιορισμοί συνδέονται με το κοντινότερο ρήμα. Όμως, οι κανόνες αυτοί δεν μπορούν να έχουν καθολική ισχύ.

Αμφιβολία στην ερμηνεία προκύπτει, επίσης, από την πολυσημία των λέξεων. Για παράδειγμα, στην παρακάτω πρόταση, η λέξη καιρός μπορεί να αναφέρεται και α) στα καιρικά φαινόμενα και β) στην έννοια του χρόνου:

Ο καιρός φταίει που οι ντομάτες δεν έχουν κοκκινίσει.

Η πιθανότητα χρήσης της λέξης με την πρώτη της σημασία είναι μεγαλύτερη και αυτή η πληροφορία πρέπει να αποτυπώνεται με κάποιο τρόπο στο λεξικό της γλώσσας. Η εξαγωγή των πιθανοτήτων υπό συνθήκη είναι εργασία των σταδίων της σημασιολογικής και της πραγματολογικής ανάλυσης.

7.9 Πραγματολογική ανάλυση

Σκοπός της **πραγματολογικής ανάλυσης** είναι η κατανόηση κειμένων και ο χειρισμός διαλόγων. Κατά την πραγματολογική ανάλυση επιχειρείται η ένταξη κάθε πρότασης μέσα στο γενικότερο νοηματικό πλαίσιο των **συμφραζόμενων** (context) της, λαμβανομένων υπόψη των συνθηκών μέσα στις οποίες αυτή λέχθηκε. Δυσκολίες προκύπτουν, όταν η πρόταση περιέχει αντωνυμίες, οι οποίες αναφέρονται σε ονόματα άλλων προτάσεων. Για παράδειγμα:

Τον αγαπάει.

Ως αίτια που οδηγούν ένα κείμενο να παρουσιάζει δυσκολίες στην κατανόησή του καταγράφονται:

- αντωνυμίες ως αντικείμενα, οι οποίες αναφέρονται σε ονόματα άλλων προτάσεων,
Ο Γιώργος είχε μια κόκκινη μπάλα. Ο Νίκος την ήθελε.
- τμήματα αντικειμένων που λείπουν,
Η Ελένη άνοιξε το βιβλίο που μόλις είχε αγοράσει. Η πρώτη σελίδα ήταν σκισμένη.
- τμήμα ενεργειών που λείπουν,
Ο Κώστας πήγε ταξίδι στην Κρήτη. Έφυγε με την πρωινή πτήση.
- οντότητες εμπλεκόμενες σε ενέργειες,
Την περασμένη εβδομάδα έγινε διάρρηξη στο σπίτι μου. Μου έκλεψαν την τηλεόραση.
- σχέσεις αιτίου – αιτιατού,
Χθες είχε φοβερή χιονοθύελλα. Τα σχολεία είναι κλειστά σήμερα.
- σειρές σχεδίων.
Η Μαρία θέλει να αγοράσει καινούριο αυτοκίνητο. Αποφάσισε να βρει δουλειά.

Για να λυθούν τα παραπάνω προβλήματα κατανόησης, πρέπει το σύστημα επεξεργασίας φυσικής γλώσσας να κατέχει τη **γνώση** για τον τρόπο με τον οποίο λειτουργεί ο κόσμος (το γενικότερο νοηματικό πλαίσιο) μέσα στον οποίο αναφέρεται η πρόταση, για τις πιθανότητες εμφάνισης κάποιων γεγονότων, για συνήθειες των ενεργούντων, πιθανά [σενάρια \(scripts\)](#) που μπορεί να διαδραματιστούν (ενδεικτικά).

Με βάση αυτήν τη γνώση και το περιεχόμενο των προτάσεων ενός κειμένου ή διαλόγου, το σύστημα μπορεί να κάνει πολλούς εύλογους συμπερασμούς, διευρύνοντας τη γνώση του για την τρέχουσα κατάσταση του κειμένου ή μιας συζήτησης. Χωρίς γενική γνώση του αντικειμένου που πραγματεύεται ένα κείμενο ή διάλογος είναι αδύνατο να γίνει πραγματολογική ανάλυση.

7.10 Παραγωγή Φυσικής Γλώσσας

Η δυνατότητα ενός συστήματος να απαντά στο χρήστη σε φυσική γλώσσα απαιτεί την **παραγωγή φυσικής γλώσσας** (natural language generation) από το σύστημα, αφού, φυσικά, κατανοήσει πρώτα το ερώτημα του χρήστη, όταν αυτό έχει επίσης εκφραστεί σε φυσική γλώσσα.

Η παραγωγή φυσικής γλώσσας πραγματοποιείται σε δύο στάδια:

- επιλογή του **τι θα λεχθεί** και
- επιλογή του **πώς αυτό θα λεχθεί**.

Το στάδιο της επιλογής του τι θα λεχθεί έχει να κάνει με το ποια πληροφορία επιλέγει το σύστημα να αναφέρει στο χρήστη. Το πρόβλημα της επιλογής της πληροφορίας αναφέρεται ως **σχεδιασμός κειμένου** (text planning) και σε εξελιγμένες περιπτώσεις δανείζεται τεχνικές από το **σχεδιασμό ενεργειών** (planning).

Για το στάδιο της επιλογής πώς θα λεχθεί μια πληροφορία, δηλαδή για τον τρόπο εμφάνισής της, η πληροφορία συνήθως ομαδοποιείται σε μικρές λογικές ενότητες, από τις οποίες δημιουργούνται προτάσεις με χρήση κανόνων γραμματικής.

Σε περίπτωση κατά την οποία η απάντηση πρέπει να εκφωνηθεί, υπάρχουν επίσης δύο προσεγγίσεις:

- να έχουν αποθηκευθεί ηχητικά όλες οι λέξεις, με όλες τις δυνατές παραλλαγές τους.
- να γίνεται σύνθεση φθόγγων από τα γράμματα των λέξεων.

Στην εκφώνηση μηχανικού προφορικού λόγου υπάρχει πάντα το πρόβλημα της διαφορετικής προφοράς μιας λέξης ανάλογα είτε με τις λέξεις που προηγούνται ή ακολουθούν είτε με τη θέση της στην πρόταση.

7.11 Αναγνώριση ομιλίας

Για να καταστεί δυνατή η **αναγνώριση ομιλίας** (speech recognition), πρέπει να υπάρχει σύστημα καταγραφής ηχητικών-ηλεκτρικών σημάτων με δυνατότητα μετατροπής τους σε φθόγγους και σύνθεσης των φθόγγων για την παραγωγή λέξεων και προτάσεων.

Τα βήματα αναγνώρισης ομιλίας είναι τρία:

13. **Παραγωγή φασματογραφήματος:** δημιουργία του φασματογραφήματος (spectrogram) του ήχου που συλλαμβάνεται από ένα μικρόφωνο.
14. **Αναγνώριση φθόγγων:** εξαγωγή των φθόγγων (phonemes) από το φασματογράφημα, βάσει μιας βιβλιοθήκης που περιέχει πρότυπα (templates) αυτών.
15. **Δημιουργία λέξεων:** συνδυασμός των φθόγγων σε λέξεις που αναγνωρίζονται με τη βοήθεια του λεξικού της γλώσσας.

7.12 Πεδία εφαρμογής Επεξεργασίας Φυσικής Γλώσσας

Η επεξεργασία φυσικής γλώσσας έχει ένα πλήθος εφαρμογών σε διάφορους χώρους. Κατά την κρίση μας, σημαντικότεροι για παρουσίαση είναι οι παρακάτω:

- **Ανάκτηση πληροφοριών** (information retrieval): χαρακτηρισμός εγγράφων με λέξεις-κλειδιά ή, στην ιδανική περίπτωση, νοηματική επεξεργασία των εγγράφων, για να δοθεί απάντηση σε ερώτηση που διατυπώνεται σε φυσική γλώσσα.
- **Ανάκτηση εγγράφων** (document retrieval).
- **Εξαγωγή πληροφορίας** (information extraction).
- **Εξόρυξη δεδομένων** (Data Mining).
- **Φωνητική προσπέλαση βάσεων δεδομένων**: Η εφαρμογή αυτή αφορά την εισαγωγή ερωτημάτων σε βάσεις δεδομένων χρησιμοποιώντας φυσική γλώσσα.
- **Κατηγοριοποίηση κειμένων** (text categorization): ταξινόμηση κειμένων βάσει του περιεχομένου τους.
- **Αυτόματη περίληψη** (automated synopsis): εξαγωγή από ένα μεγάλο κείμενο ενός μικρότερου, με το βασικό/κεντρικό νόημα του πρώτου. Επιτρέπει την παραγωγή περιλήψεων από τα επιστρεφόμενα έγγραφα μιας μηχανής αναζήτησης.
- **Αυτόματη μηχανική μετάφραση/διερμηνεία** (machine translation).
- **Αναγνώριση μερών του λόγου** (part-of-speech tagging),
- **Συντακτική ανάλυση** (parsing),
- **Εφαρμογές Επεξεργασίας Κειμένου** (Word Processing Applications): έλεγχος ορθογραφίας, εισαγωγή κειμένου.
- **Εκμάθηση γλώσσας υποβοηθούμενη από Η/Υ**,
- **Κατασκευή ηλεκτρονικών λεξικών**: Επέκταση υπάρχοντων λεξικών π.χ λεξικό ιατρικής ορολογίας (Wolff, 1984).

Παρά την πληθώρα τους, οι εφαρμογές στις οποίες απαιτείται κατανόηση φυσικής γλώσσας συχνά αποτυγχάνουν στο να κατανοήσουν και να διατυπώσουν σύνθετες ερωτήσεις. Για να υπάρχει ικανοποιητικό αποτέλεσμα, ο χρήστης θα πρέπει να γνωρίζει το υποστηριζόμενο λεξιλόγιο και τις γραμματικές/συντακτικές δυνατότητες του συστήματος. Ίσως και για το λόγο αυτό δε γνώρισαν έως τώρα μεγάλη εμπορική επιτυχία.

Βιβλιογραφία/Αναφορές

- Jurafsky, D. & Martin, J. H. (2000). *Speech and Language Processing*. Νιου Τζέρσεϊ: Prentice-Hall. Ανακτήθηκε από http://www.deepsky.com/~merovech/voynich/voynich_manchu_reference_materials/PDFs/jurafsky_martin.pdf
- Jurafsky, D. & Martin, J. H. (2009). *Speech and Language Processing* (2^η έκδοση). Νιου Τζέρσεϊ: Prentice-Hall.
- Montague, R. (1970a). English as a formal language. Στο B. Visentini κ.ά., (εκδ.), *Linguaggi nella societa e nella tecnica* (σ.189–224). Μιλάνο: Edizioni di Communita.
- Montague, R. (1970b). Universal grammar. *Theoria*, 36, 373–398.

- Montague, R. (1973). The proper treatment of quantification in ordinary English. Στο J. Hintikka, J. M. E. Moravcsik και P. Suppes (εκδ.), *Approaches to Natural Language* (σ. 221–242). Dordrecht: D. Reidel.
- Schank, R.C. (1969). *A Conceptual Dependency Representation for a Computer-Oriented Semantics* (Διδακτορική Διατριβή), Πανεπιστήμιο του Τέξας, ΗΠΑ.
- Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding: An inquiry into human knowledge structures*. Hillsdale, Νιου Τζέρσεϊ : Erlbaum Assoc.
- Thomason, R.H. (εκδ.).(1974). *Formal Philosophy: Selected Papers of Richard Montague*. New Haven, Κοννέκτικατ: Yale University Press.
- Weizenbaum, J. (1966). ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM*, 9(1), 36-45.
- Winograd, T. (1972), *Understanding Natural Language*. Νέα Υόρκη: Academic Press. Την ίδια χρονιά και έκδοση στο *Cognitive Psychology*, 3(1), 1-191.
- Wolff, S. (1984). The use of morphosemantic regularities in the medical vocabulary for automatic lexical coding. *Methods of Information in Medicine*, 23,195 -203.
- Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Ρεφανίδης, Ι., Κοκκοράς, Φ. & Σακελλαρίου, Η. (2011). *Τεχνητή Νοημοσύνη* (3^η έκδοση). Θεσσαλονίκη: Εκδόσεις Πανεπιστήμιου Μακεδονίας.

Κριτήρια αξιολόγησης

Κριτήριο αξιολόγησης 1

Σε ποιους τομείς μπορεί να είναι ασαφής μια πρόταση φυσικής γλώσσας;

5. λεξιλογικό,
6. προτασιακό,
7. συντακτικό,
8. συμβολικό,
9. αναφορικό,
10. σημασιολογικό,
11. πραγματολογικό.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 1^η, 3^η, 5^η, 6η και η 7^η.

Κριτήριο αξιολόγησης 2

Από τι αποτελούνται τα συστήματα γραμματικής;

1. δομή,
2. σύνταξη,
3. γραμματική,
4. συντακτικό αναλυτή,
5. παραγωγή κώδικα.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 3^η, και η 4^η.

Κριτήριο αξιολόγησης 3

Τι από τα παρακάτω κάνει ένας συντακτικός αναλυτής;

1. Διορθώνει τα προτασιακά λάθη.
2. Εντοπίζει την ασάφεια σε μια λέξη.
3. Ελέγχει τη συντακτική ορθότητα μιας πρότασης.
4. Δημιουργεί το συντακτικό δένδρο.
5. Δημιουργεί κώδικα μετάφρασης.

Απάντηση/Λύση

Σωστές απαντήσεις είναι η 3^η, και η 4^η.

Ευρετήριο Όρων

A

αβεβαιότητα (uncertainty), 206
Ακτινωτή μέθοδος αναζήτησης (Beam search), 61
αλγόριθμος
αναζήτησης (search algorithm), 24, 36
αναζήτησης με απαγορεύσεις (Taboo Search Algorithm), 60
εκπαίδευσης ΤΝΔ (ANN training algorithm), 167
εξελκτικός (evolutionary algorithm), 179
επαγωγικής μάθησης (inductive learning algorithm), 130
ID3 (Iterative Dichotomizer 3), 135
προσομοιωμένης απόπτωσης (Simulated Annealing Algorithm), 60
αναγνώριση ομιλίας (speech recognition), 255
αναζήτηση (search), 24
Άλφα-Άστρο (A*), 63
Αναρρίχησης Λόφων (Hill-Climbing - HC), 57
Βασικής Επέκτασης και Οριοθέτησης (Basic Branch & Bound search - B&B), 60
Διπλής Κατεύθυνσης (Bidirectional Search), 52
Εξαναγκασμένη Αναρρίχηση Λόφων (Enforced Hill Climbing - EHC), 59
εξαντλητική (exhaustive search), 38
Επαναληπτικής Εμβάθυνσης (Iterative Deepening Search - IDS), 51
ευρετική αναζήτηση (heuristic search), 53
Πρώτα σε Βάθος (Depth-First Search - DFS), 45
Πρώτα σε Πλάτος (Breadth-First Search - BFS), 48
Πρώτα στο Καλύτερο ή Καλύτερη-Πρώτη (Best First Search - BestFS), 55
αναπαραγωγή (reproduction) ΓΑ, 185
αναπαράσταση (representation), 71
αναπαράσταση της γνώσης (knowledge representation), 74
ανατροφοδοτούμενα ΤΝΔ (recurrent ANN), 160
ανοιχτός κόσμος (open world), 27
αντιδραστικοί πράκτορες (reactive agents), 225, 226
αντιδραστικός πράκτορας βασισμένος σε μοντέλο (model-based reactive agent), 227
άξονας του κυττάρου-νευροάξονα (axon), 156
άπληστη τοπική αναζήτηση (greedy local search), 58
απλός αντιδραστικός πράκτορας (simple reactive agent), 227
απόκτηση γνώσης (knowledge acquisition), 201 205
αρχική κατάσταση (start state ή input state), 27
ασάφεια (fuzziness), 209
σε αναφορικό επίπεδο (ambiguity at referential level), 243
σε επίπεδο λεξιλογικό (ambiguity at lexical level), 243

σε επίπεδο σύνταξης (ambiguity at syntactic level), 242
σε πραγματολογικό επίπεδο (pragmatic level), 243
σε σημασιολογικό επίπεδο (ambiguity at semantic level), 243
στη γλώσσα (ambiguity of language), 242
ασαφείς κανόνες (fuzzy rules), 209, 211
ασαφή σύνολα (fuzzy sets), 209
ασαφής λογική (fuzzy logic), 210
άτομα (atoms), 77
αυξητική εκπαίδευση (incremental training), 171
αυτόματο πεπερασμένων καταστάσεων (finite state automaton-FSA), 250
αυτο-οργανωνόμενοι χάρτες (self-organizing maps), 172
αυτοσυσχετιζόμενες μνήμες (autoassociated memories), 160

B

βάθος μονοπατιού (path depth), 39
Βάση Γνώσης (Knowledge Base), 108 201
Βάση Δεδομένων (DataBase), 201
Βάση Κανόνων (RuleBase), 108 201
βασικός χώρος επίλυσης προβλήματος (basic problem solution space), 123
βηματικές συναρτήσεις ή συναρτήσεις κατωφλίου (threshold functions), 161
βιολογικοί πράκτορες (biological agents), 222

Γ

γεγονός (fact), 108
γενεαλογική αντικατάσταση (generational replacement), 194
γενετική ποικιλομορφία (generational diversity), 195
Γενετικοί Αλγόριθμοι-ΓΑ (Genetic Algorithms-GA), 178
γενετικοί τελεστές (genetic operators), 189
Γενετικός Προγραμματισμός (Genetic Programming), 196
γενική γνώση (general knowledge, common sense), 73
Γενικός Επιλυτής Προβλημάτων - ΓΕΠ (General Problem Solver - GPS), 123
Γκαουσιανές συναρτήσεις (Gaussian functions), 162
γνωρίσματα (attributes), 130
γνώση (knowledge), 71
Γνωστική ή Γνωσιακή Επιστήμη (Cognitive Science), 14
γονίδια (genes), 178
γονότυπος (genotype), 182
Γραμματικές Οριστικών Προτάσεων-ΓΟΠ (Definite Clause Grammars-DCGs), 249

γραμματική (grammar), 247
γραμματική ανάλυση πρότασης (parsing), 247

Δ

δεδομένα (data), 71
δεδομένων ελέγχου (test data), 130
Δένδρα Απόφασης-ΔΑ (Decision Trees), 131
δενδρίτες (dendrites), 156
δένδρο αναζήτησης (search tree), 36
 απόφασης (decision tree), 134
δηλωτικές προτάσεις (declarative sentences), 76
δηλωτική γνώση (declarative knowledge), 72
διαδικαστική γνώση (procedural knowledge), 72
διάδοχοι κόμβοι (successor nodes), 39
διάδοχος κατάσταση (successor state), 27
διαπραγμάτευση (negotiation), 237
διασταύρωση (crossover), 190
 δύο σημείων (two-points crossover), 191
διατύπωση προβλήματος (problem formulation), 26
διαχειριστής έργου (project manager), 205
Διερμηνευτική ΣΒΠ/CBR (Interpretive CBR), 154
δίκτυο Hopfield, 174

Ε

εγγύτητα (closeness), 252
ειδική γνώση (domain-specific knowledge), 73
ειδικός τομέα (domain expert), 204
Ελιτισμός (Elitism), 195
Έμπειρο Σύστημα-ΕΣ (Expert System - ES), 200
Έμπειρο Σύστημα Βασισμένο σε Κανόνες-ΕΣΒΚ
 (Rule-Based Expert System-RBES), 201
ενεργοί κανόνες (active rules), 95
ενεργοποιητής (activator), 226
ενεργός μνήμη (active memory), 108
ενισχυτική μάθηση (reinforcement learning), 128
ενοιολογικός χάρτης- EX (concept map), 85
ενοποίηση (unification), 106
ενός σημείου (single point crossover), 190
εντροπία πληροφορίας (information entropy), 136
Εξαναγκασμένη Αναρρίχηση Λόφων (Enforced Hill
 Climbing – EHC), 59
εξαντλητική αναζήτηση (exhaustive search), 38
εξελικτική μάθηση (evolutionary learning), 128, 178
Εξελικτική Υπολογιστική (Evolutionary Computation),
 179
εξελικτικοί αλγόριθμοι (evolutionary algorithms), 179
εξελικτικός κύκλος (evolutionary cycle), 179
επαγωγή (induction), 102, 127
επαγωγική μάθηση (inductive learning), 127
επεισοδιακή γνώση (episodical knowledge), 72
επεξεργασία φυσικής γλώσσας (natural language
 processing-NLP), 242
επιβλεπόμενη Επαγωγική Μάθηση (supervised
 Inductive Learning), 130
επιβλεπόμενη μάθηση (supervised learning), 128
επικοινωνία ανθρώπου-μηχανής (human-computer
 interaction), 242
επιλεκτική πίεση (selective pressure), 195

επιλογή τουρνουά (tournament selection), 188
επίλυση σύγκρουσης (conflict resolution), 113
επίπεδο (layer), 158
επίπεδο Kohonen (Kohonen layer), 171
εποχή εκπαίδευσης (training epoch), 171
ετεροαρχικό σημασιολογικό δίκτυο (non-hierarchical
 neural network), 83
ετεροσυσχετιζόμενες μνήμες (heteroassociated
 memories), 160
ευρετική αναζήτηση (heuristic search), 53
ευρετική τιμή (heuristic value), 54
ευρετικός μηχανισμός (heuristic), 53
ευφυείς διεπαφές χρήσης (intelligent intuitive
 interfaces), 20
ευφυείς πράκτορες (intelligent agents), 217
ευφυές έμπειρο σύστημα (intelligent expert system),
 205
ευφυΐα (intelligence), 69

Η

ηχητικό μοντέλο (sound-based model), 253

Ι

ιεραρχικό σημασιολογικό δίκτυο (hierarchical semantic
 network), 81
ιοί (viruses), 223

Κ

κακοσχηματισμένες προτάσεις (ill formed sentences),
 246
καλοσχηματισμένες προτάσεις (well formed sentences),
 246
καλοσχηματισμένοι τύποι (well-formed formulas), 77
 78
κανόνας (rule), 94
 Δέλτα (Delta rule), 168
 εκμάθησης (learning rule), 168
 παραγωγής (productive rule), 95
 παραγωγής ή παραγωγικοί κανόνες (production
 rule), 111
 συμπερασματικός (inference rule), 109
 συνεπαγωγικός (deductive rule), 95
 συντακτικός κανόνες (syntax rule), 248
 ταξινόμησης (classification rule), 132
 του Hebb (Hebbian rule), 168
κανονικό γενετικό αλγόριθμο (canonical genetic
 algorithm), 182
κατανόηση φυσικής γλώσσας (natural language
 understanding-NLU), 242
κατάσταση προβλήματος (problem state), 25
κατάσταση-στόχος (goal state), 27
κατηγορήματα (predicates), 78
κατηγορηματικός λογισμός (predicate calculus), 77
κατηγορηματικός λογισμός πρώτης τάξης (first-order
 predicate calculus), 78
κατηγορίες ταξινόμησης (classes), 132
κέλυφος έμπειρου συστήματος (expert system shell,
 203

κέρδος πληροφορίας (information gain), 138
κινητοί πρακτόρες (mobile agents), 233
κλειστό σύνολο (closed set), 40
κλειστός κόσμος (closed world), 27
κόσμος προβλήματος (problem world), 26
κόστος μονοπατιού (path cost), 39
κρυμμένα επίπεδα ΤΝΔ (hidden layers), 158
κυτταρικό σώμα (cell body), 156

Λ

λεξικό (lexicon), 248
λογικοί σύνδεσμοι (logical connectives), 78
λογισμικοί πράκτορες (software agents), 222
λύση προβλήματος (problem solution), 34, 35

Μ

μαζική εκπαίδευση (batch training), 171
μάθηση (learning), 127
επαγωγική μάθηση (inductive learning), 127
με οπισθοδιάδοση σφάλματος (error back-propagation), 169
μη επιβλεπόμενη (unsupervised learning), 128
μερική λύση προβλήματος (partial problem solution), 35
μεταβλητή στόχου (goal variable), 130
μετάλλαξη (mutation), 192
scrabble lists (scrabble lists mutation), 193
τυχαίας ανταλλαγής, 193
μέτρο εμπιστοσύνης (measure of belief), 207
μέτωπο (fringe ή front), 40
μήκος ορισμού (defining length) ενός σχήματος, 184
μη-τερματικά σύμβολα (non-terminals), 248
μηχανική της γνώσης (knowledge engineering), 205
Μηχανική Μάθηση (Machine Learning), 127
μηχανική όραση (machine vision), 13
μηχανικός γνώσης (knowledge engineer), 204
μηχανισμός εξαγωγής συμπερασμάτων (inference mechanism), 108 201
μνήμη
εργασίας (working memory), 108 201
μακράς διάρκειας (long-term memory), 201
μικρής διάρκειας (short-term memory), 201
μονάδες ΤΝΔ (units) ή κόμβοι (nodes), 158
μονοπάτι αναζήτησης (search path), 36
μοντέλο, 104
γνώσης (knowledge model), 87
συστήματος παραγωγής (production system model), 201
χαρακτήρων (letter-based model), 252
μορφολογική ανάλυση (morphological analysis), 246

Ν

νευρώνες (neurons), 155
Νευρωνικά Δίκτυα (Neural Networks, Connectionist Networks, Parallel Distributed Processing Models), 155
νοητικά πρότυπα (mental patterns), 127
νοητικό μοντέλο (mental model), 127

νοητικός χάρτης (mind map), 86

Ο

ο στόχος (goal), 27
ομοιόμορφη διασταύρωση (uniform crossover), 191
ομοιόμορφη μετάλλαξη (uniform mutation), 193
οντολογία (ontology), 96
οντολογίες (ontologies), 247
οπίσθια τροφοδότηση (feed backward), 159
ορθολογικοί πράκτορες (rational agents), 223
Ορισμός Προβλήματος, 26
ουρά αναζήτησης (search queue ή fringe), 36

Π

παραγωγή (deduction), 104
παραγωγή φυσικής γλώσσας (natural language generation), 255
παραγωγική μνήμη (production memory), 111 201
πατρικός ή γονικός κόμβος (parent node), 39
Περιρρέουσα Νοημοσύνη (Ambient Intelligence-Aml), 20
πλαίσια (frames), 88
πληροφορία (information), 71
πολυπρακτορικό σύστημα (multi-agent system), 233
πολυσημία (ambiguity), 253
πραγματολογική ανάλυση, 254
πράκτορες (agents), 217
βασισμένοι στη λογική (deliberated agents), 228
επιμέρους εργασιών (task-specific agents), 223
με εσωτερική κατάσταση (internal state agents), 225, 228
με πεποιθήσεις - επιθυμίες - προθέσεις (BDI agents), 228, 229
με στόχους (goal-based agents), 228
ρομποτικοί (robotic agents ή robots), 222
τεχνητής ζωής (artificial life agents), 223
τεχνητοί (artificial agents), 222
υβριδικοί πρακτόρων (hybrid agents), 232
ψυχαγωγίας (entertainment agents), 223
προγραμματιστής (programmer), 205
προς τα εμπρός αλυσιδωτή εκτέλεση κανόνων (forward chaining), 113
προς τα πίσω αλυσιδωτή εκτέλεση κανόνων (backward chaining), 119
πρόσθια τροφοδότηση (feed forward), 159
προτασιακός λογισμός (propositional calculus), 76
πρότυπο (pattern), 106
πρότυπο ομοιότητας (prototype resemblance), 184
Πρώτα σε Βάθος Αναζήτηση (Depth-First Search - DFS), 45
Πρώτα σε Πλάτος Αναζήτηση (Breadth-First Search - BFS), 48
πυροδότηση κανόνα (rule firing), 108

Ρ

ρομποτικής (robotics), 13
ρομποτικοί πράκτορες (robotic agents ή robots), 222

ρουλέτα γονικής επιλογής (roulette wheel selection), 185

Σ

σενάριο (script), 92
σημασιολογικά δίκτυα (semantic networks), 80
σημασιολογική ανάλυση (semantic analysis), 253
σημασιολογική γνώση (semantic knowledge), 72
σιγμοειδείς συναρτήσεις (sigmoid functions), 162
σοφία (wisdom), 71
στιγμιότυπο (instance), 130
στοιχειώδης Perceptron (basic Perceptron), 158
στόχος (goal), 228
στόχος προβλήματος (problem goal), 25
στρατηγική επίλυσης συγκρούσεων (conflict resolution strategy), 112, 113
συγκέντρωση γνώσης (knowledge elicitation), 205
συλλογιστική (reasoning), 102
 Βασισμένη σε Περιπτώσεις-ΣΒΠ (Case-Based Reasoning-CBR), 149
 με Αναλογίες (Analogical Reasoning), 149
σύμβολα (symbols), 77
συμβολική Τεχνητή Νοημοσύνη (symbolic Artificial Intelligence):, 14
συμπερασματική μηχανή (inference engine), 108 201
συμπερασματικοί κανόνες (inference rules), 109
συνάρτηση
 αθροίσματος (summation function), 157
 αναρρίχησης (ramping function), 161
 βηματικής μεταβολής (hard limiter function), 161
 διαδόχων (successor function), 27
 καταλληλότητας (fitness function), 181
 κόστους διαδρομής (function of path cost), 28
 λάθους (error function), 130
 μετάβασης (transfer functions), 157 161
 πρόγνωσης (predictor function), 130
 προσήμου (sign function), 161
 στόχου (goal function), 130
 συμμετοχής (membership function), 210
 χρησιμότητας (utility function), 230
συναρτησιακά σύμβολα (functional symbols), 77
σύναψη (synapse), 156
συνεπαγωγικοί κανόνες (deductive rules), 95
συνεργασία (cooperation), 237
σύνολο
 δεδομένων εκπαίδευσης (training set), 129
 εκπαίδευσης (training set), 130
 ελέγχου (test set), 130
 σύγκρουσης (conflict set), 112
συντακτική ανάλυση (syntactic analysis), 247
συντακτικό δένδρο (parse tree), 249
συντακτικοί κανόνες (syntax rules), 248
συντακτικός αναλυτής (parser), 249
συντελεστές βεβαιότητας (certainty factors-CF), 207
συστήματα
 ανταλλαγής μηνυμάτων (message passing systems), 235, 236
 Βασισμένα σε Κανόνες (Rule-Based Systems-ΣBK/RBS), 95 105 107

Βασισμένα στη Γνώση - ΣΒΓ (Knowledge-Based Systems - KBS), 105 200

Εξαγωγής Συμπερασμάτων (Deduction Systems), 107 108

μαυροπίνακα (blackboard systems), 235

Παραγωγής (Production Systems), 107 111

συσχετιστική μνήμη (autoassociative memory), 174

σχεδιασμός ενεργειών (planning), 255

σχεδιασμός κειμένου (text planning), 255

σχέση συνεπαγωγής (implication relation), 211

σχήμα (schema), 88, 184

σχισμές (slots), 89

T

ταίριασμα προτύπων (pattern matching), 106

τάξη (order) ενός σχήματος, 184

τελεστής μετάβασης (transition operator), 27

τελικός χρήστης (end user), 205

Τερματικά σύμβολα (terminals), 248

τεστ Τούρινγκ (Turing test), 12

Τεχνητά Νευρωνικά Δίκτυα-TΝΔ (Artificial Neural Networks-ANN), 157

 Backpropagation (Backpropagation ANN), 169

 Kohonen (Kohonen ANN), 171

 Perceptron, 164

Τεχνητή Νοημοσύνη-TN (Artificial Intelligence-AI), 12

τεχνητοί πράκτορες (artificial agents), 222

τιμή αλήθειας (truth value), 76

τιμής κατωφλίου (threshold value), 157

Τμήμα διεπαφής (Interface), 201

Τμήμα Επεξηγήσεων (Explanations Facilities), 201

τυφλές μέθοδοι αναζήτησης (blind search methods), 45

Υ

υβριδικοί πρακτόρων (hybrid agents), 232

Υπολογιστική Νοημοσύνη-ΥΝ (Computational Intelligence-CI), 15

υπολογιστικοί πράκτορες (computational agents), 222

Φ

φαινότυπος (phenotype), 183

X

χρονική οπισθοδρόμηση (chronological backtracking), 47

χρωμοσώματα (chromosomes), 178

χώρος

 αναζήτησης (search space), 35

 καταστάσεων (state space), 25, 27

