

Εισαγωγή

Διδάσκων:

Δρ. Δημήτρης Αποστόλου, Καθηγητής, Τηλ: 4142476, Email: dapost στο unipi.gr

Δικτυακός χώρος μαθήματος: <https://gunet2.cs.unipi.gr/courses/TMB109/>

Γραπτή εξέταση και προαιρετική εργασία με bonus 2 βαθμούς

Βιβλίο 1: Τεχνητή Νοημοσύνη - Δ' Έκδοση

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου

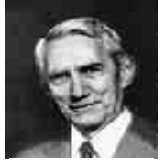
Βιβλίο 2: Τεχνητή Νοημοσύνη – Μία σύγχρονη προσέγγιση - Δ' Έκδοση

S. Russell, P. Norvig (Απόδοση στα Ελληνικά: Γ. Ρεφανίδης)



John

- ❖ Η ΤΝ συμπλήρωσε 60 χρόνια ζωής.
- ❑ Είναι ένα από τα πιο νέα ερευνητικά πεδία.
- ❑ Τυπικά ξεκίνησε το 1956 στη συνάντηση μερικών επιφανών επιστημόνων, όπως ο John McCarthy, ο Marvin Minsky, ο Claude Shannon, κ.α.



Claude
Shannon

- ❖ Η μελέτη της νοημοσύνης είναι ένα από τα πιο παλιά θέματα.

- ❖ Το μάθημα προσεγγίζει το θέμα της Τ.Ν.:

- ❑ Οριοθετώντας το πρόβλημα, παρουσιάζοντας αλγορίθμους αναζήτησης και περιγράφοντας τρόπους αναπαράστασης, προσπαθώντας δηλαδή να επιλύσει την εξίσωση: Τ.Ν. = Αναπαράσταση Γνώσης + Αναζήτηση.

- ❑ Συμπληρώνεται με τη μελέτη κλασσικών εφαρμογών που ιστορικά ανήκουν στην περιοχή, όπως: τα έμπειρα συστήματα, τους γενετικούς αλγορίθμους, κ.α.

- ❑ Για λόγους πληρότητας παρουσιάζονται συνοπτικά και άλλες δημοφιλείς εφαρμογές όπως: οι πράκτορες, η ρομποτική, ο σημασιολογικός ιστός, κ.α.

Τι είναι Τεχνητή Νοημοσύνη

ΤΝ είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζουμε με τη νοημοσύνη στην ανθρώπινη συμπεριφορά (Barr και Feigenbaum).

- ❖ Πριν ορίσουμε την Τεχνητή Νοημοσύνη πρέπει να ορισθεί πρώτα η νοημοσύνη.

Τι είναι Νοημοσύνη

- ❖ Ο Howard Gardner στο βιβλίο του *Frames of Mind: The theory of multiple intelligences* (1983), διακρίνει σε κάθε άνθρωπο 8 τύπους νοημοσύνης (Γλωσσική, Λογική/Μαθηματική, Μουσική, Χωρική, Σωματική, Διαπροσωπική, Ενδοπροσωπική, Φυσιοκρατική) οι οποίοι είναι ευδιάκριτοι μέσα στον εγκέφαλο αλλά στην πράξη χρησιμοποιείται ένα μίγμα από αυτούς.
- ❖ Στο ερμηνευτικό λεξικό του Cambridge (<http://dictionary.cambridge.org/>) θα βρούμε ότι νοημοσύνη είναι η ικανότητα για μάθηση, κατανόηση και κρίση ή αιτιολογημένη έκφραση γνώμης.
- ❖ Στο λεξικό Merriam-Webster (<http://www.m-w.com/>) αναφέρεται ότι νοημοσύνη είναι η ικανότητα για μάθηση ή κατανόηση ή η αντιμετώπιση νέων ή δύσκολων καταστάσεων.
- ❖ Χαρακτηριστικά / εκφάνσεις νοημοσύνης:
 - Δυνατότητα λόγου
 - Δυνατότητα συλλογισμών
 - Δυνατότητα μάθησης

Ορισμός της Νοημοσύνης

- ❖ Ο Douglas Hofstadter (βραβείο Pulitzer), προτείνει ότι νοημοσύνη είναι, να:
 - ❑ Ανταποκρίνεσαι σε καταστάσεις με ελαστικότητα (όχι μηχανική συμπεριφορά).
 - ❑ Κατανοείς τα ασαφή ή αντιφατικά μηνύματα από τα συμφραζόμενα.
 - ❑ Αναγνωρίζεις και να ιεραρχείς τα διάφορα δεδομένα με βάση τη σπουδαιότητα τους.
 - ❑ Βρίσκεις ομοιότητες μεταξύ καταστάσεων οι οποίες μοιάζουν διαφορετικές.
 - ❑ Βρίσκεις διαφορές μεταξύ καταστάσεων οι οποίες μοιάζουν παρόμοιες.

- ❖ Οι ικανότητες αυτές έχουν τουλάχιστον ένα κοινό χαρακτηριστικό: αποκτώνται εύκολα από τους ανθρώπους και βασίζονται συνήθως σε ένα σύνολο σταθερών και στερεότυπων απόψεων/γνώσεων που κατέχει οποιοσδήποτε άνθρωπος και αποκαλείται *κοινή λογική (common sense)*.

Ορισμός της TN

❖ Άλλοι ορισμοί επικεντρώνονται στη διαδικασία σκέψης και συλλογισμού και άλλοι στη συμπεριφορά. Υπάρχουν ορισμοί σύμφωνα με τους οποίους στόχος της TN είναι να φτιάξει συστήματα που:

- Σκέφτονται όπως οι άνθρωποι
- Συμπεριφέρονται όπως οι άνθρωποι
- Σκέφτονται λογικά
- Αντιδρούν λογικά

Γενικός Ορισμός TN

TN είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κλπ.

Περιοχές της ΤΝ

- ❑ Επίλυση προβλημάτων
- ❑ Απόδειξη Θεωρημάτων
- ❑ Επεξεργασία Φυσικής Γλώσσας
- ❑ Τεχνητή Όραση
- ❑ Μηχανική Μάθηση
- ❑ Σχεδιασμός Ενεργειών και Χρονοπρογραμματισμός
- ❑ Αυτόνομα Robot,
- ❑ Έμπειρα Συστήματα και Συστήματα Γνώσης
- ❑ Ευφυείς πράκτορες (agents)
- ❑ Ευφυείς υπηρεσίες διαδικτύου και σημασιολογικό διαδίκτυο (semantic web)
- ❑ Προσαρμοζόμενα και εξελισσόμενα ευφυή συστήματα
- ❑ και άλλες

Προσεγγίσεις για την ΤΝ

- ❖ Κλασική ή συμβολική (symbolic AI): Βασίζεται στην κατανόηση των νοητικών διεργασιών και ασχολείται με τη προσομοίωση της ανθρώπινης νοημοσύνης προσεγγίζοντάς την με αλγορίθμους και συστήματα που βασίζονται στη γνώση χρησιμοποιώντας ως δομικές μονάδες τα σύμβολα (π.χ. συστήματα κανόνων).
- ❖ Υπολογιστική νοημοσύνη (computational intelligence) ή Συνδετική (connectionist) ή μη-συμβολική: Βασίζεται στη μίμηση της βιολογικής λειτουργίας του εγκεφάλου όπως η διαδικασία της εξέλιξης των ειδών ή η λειτουργία του εγκεφάλου (π.χ. νευρωνικά δίκτυα, γενετικοί αλγόριθμοι).

Δοκιμασία Turing

❖ Ο Alan Turing (1913-1954) ο οποίος θεωρείται ο πατέρας της ΤΝ, εμπνεύστηκε το 1950 ένα τεστ (Turing test), για την αναγνώριση ευφυών μηχανών.



Alan Turing

- ❑ Το Turing test βασίζεται σε μία σειρά από ερωτήσεις που υποβάλει κάποιος σε έναν άνθρωπο και μία μηχανή, χωρίς να ξέρει εκ των προτέρων ποιος είναι ποιος.
- ❑ Αν στο τέλος δεν καταφέρει να ξεχωρίσει τον άνθρωπο από τη μηχανή, τότε η μηχανή περνάει το τεστ και θεωρείται ευφυής.

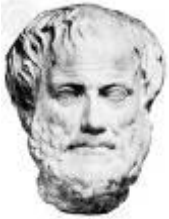
❖ Ο προγραμματισμός ενός υπολογιστή για να περάσει το τεστ, απαιτεί τη συμμετοχή αρκετών επιστημονικών πεδίων, όπως:

- ❑ Επεξεργασία φυσικής γλώσσας (Natural Language Processing, NLP).
- ❑ Αναπαράσταση γνώσης.
- ❑ Αυτοματοποιημένη συλλογιστική.
- ❑ Μηχανική μάθηση.

❖ Μια επέκταση του τεστ (**πλήρες Turing τεστ**) περιλαμβάνει και την αναγνώριση εικόνων και αντικειμένων.

- ❑ Απαιτείται η συμμετοχή και άλλων δύο επιστημονικών πεδίων, της μηχανικής όρασης (machine vision) και της ρομποτικής (robotics).

Η Εξέλιξη της ΤΝ



- ❖ Οι "συλλογισμοί" του Αριστοτέλη (384-322 π.Χ.) παρείχαν πρότυπα εκφράσεων που έδιναν πάντα σωστά συμπεράσματα από σωστές υποθέσεις.
- ❖ 1854: Ο George Boole έθεσε τις βάσεις της προτασιακής λογικής.



George Boole

- ❖ 1879: Ο Gottlieb Frege πρότεινε ένα σύστημα αυτοματοποιημένης συλλογιστικής και έθεσε τις βάσεις του κατηγορηματικού λογισμού (predicate calculus).
- ❖ 1943: Ο McCulloch και ο Pitts πρότειναν ένα μοντέλο τεχνητών νευρώνων που είχε τη δυνατότητα να μαθαίνει και να υπολογίζει κάθε υπολογίσιμη συνάρτηση.
- ❖ 1951: Ο Minsky και ο Edmouts υλοποίησαν το πρώτο νευρωνικό δίκτυο, το SNARC, με 40 νευρώνες, το οποίο χρησιμοποιούσε 3.000 λυχνίες.
- ❖ 1956: Διοργάνωση συνεδρίου (workshop) καθοριστικού στη γέννηση της Τ.Ν.

Η Εξέλιξη της ΤΝ

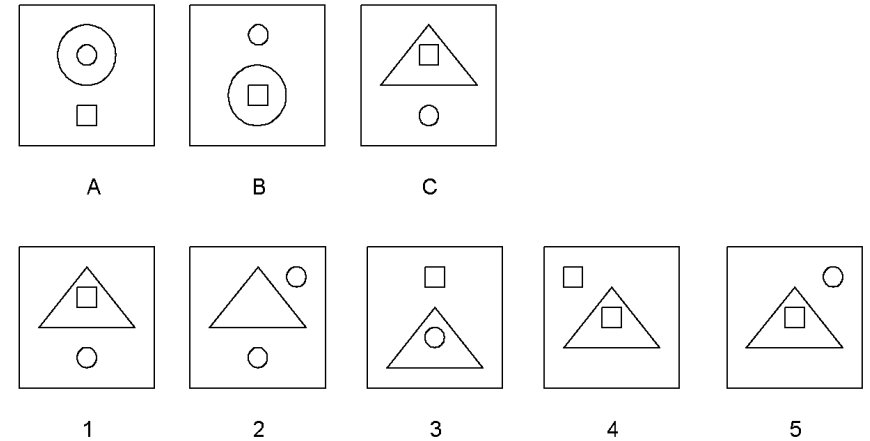
Δεκαετία '60

- ❖ 1958: Ο McCarthy:
 - ❑ Όρισε τη συναρτησιακή γλώσσα LISP.
 - ❑ Πρότεινε ένα υποθετικό σύστημα (τον advice taker), που χρησιμοποιούσε γνώση (όπως το LT) αλλά αφορούσε γενικά, καθημερινά, προβλήματα.
- ❖ 1958: Ο Friedberg πρότεινε μια τεχνική, τη μηχανική εξέλιξη (machine evolution) ή όπως ονομάζεται τώρα, γενετικοί αλγόριθμοι (genetic algorithms).
- ❖ Δεκαετία του '60:
 - ❑ Στο Stanford υλοποιήθηκε το πρώτο robot, το Shakey robot.

□ 1968: Το πρόγραμμα ANALOGY του Tom Evans έλυσε προβλήματα γεωμετρικής αναλογίας που χρησιμοποιούνταν σε τεστ ευφυΐας.

□ 1962: Βελτιώσεις της μεθόδου μάθησης των νευρωνικών δικτύων του Hebb από τον Rosenblatt με τα perceptrons.

Αν το A αντιστοιχεί στο B τότε το C σε ποιο από τα ακόλουθα γεωμετρικά σχήματα αντιστοιχεί;



Η Εξέλιξη της ΤΝ

Δεκαετία '70

- ❖ Τον ενθουσιασμό της πρώτης δεκαετίας της Τ.Ν. διαδέχθηκε η εποχή της κριτικής ότι τα συστήματα ήταν κατάλληλα μόνο για παιχνίδια (toy problems).
- ❖ Τη δεκαετία του '70 αναπτύχθηκαν συστήματα που περιείχαν την απαιτούμενη γνώση ώστε να συμπεριφέρονται όπως οι άνθρωποι ειδικοί σε διάφορα θέματα.
- ❑ Ονομάστηκαν Έμπειρα Συστήματα (**Expert Systems**) ή Συστήματα Γνώσης (**Knowledge Systems**):
 - ❑ Αρχές δεκαετίας του '70: Προτάθηκε η γλώσσα προγραμματισμού Prolog.
 - ❑ 1975: Προτάθηκαν από τον Minsky τα πλαίσια (frames).

Δεκαετία '80

- ❖ Το 1981 οι Ιάπωνες ανακοίνωσαν το πρόγραμμα της 5^{ης} γενιάς, ένα δεκαετές πρόγραμμα για την κατασκευή υπολογιστών με γλώσσα μηχανής την Prolog.
- ❑ Στόχος ήταν να κατασκευαστούν ευφυή συστήματα, τα οποία εκτός των άλλων, θα ήταν σε θέση να επικοινωνούν πλήρως με τον άνθρωπο σε φυσική γλώσσα.
- ❖ Στα μέσα της δεκαετίας του '80 επανεμφανίστηκαν τα νευρωνικά δίκτυα.

Η Εξέλιξη της ΤΝ

Περίοδοι

- ❖ Πολλοί συγγραφείς διακρίνουν στην ιστορία της ΤΝ τέσσερις περιόδους.
 - ❑ Προϊστορική: η ΤΝ ουσιαστικά προαναγγέλλεται σε διηγήματα επιστημονικής φαντασίας.
 - ❑ Κλασική (μέχρι τα μέσα της δεκαετίας του 1960): αναπτύχθηκαν συστήματα που έπαιζαν παιχνίδια και έλυναν γρίφους.
 - ❑ Ρομαντική (μέχρι τα μέσα της δεκαετίας του 1970): οι προσπάθειες επικεντρώνονται στην ανάπτυξη συστημάτων που κατανοούν ιστορίες και διάλογους σε φυσική γλώσσα.
 - ❑ Μοντέρνα (μέχρι τα τέλη της δεκαετίας του 1980): χαρακτηρίζεται από την ανάπτυξη συστημάτων που βασίζονται στη γνώση και την εμπορική εκμετάλλευση των αποτελεσμάτων της έρευνας γύρω από την ΤΝ.

- ❖ Αυτήν την εποχή βιώνουμε τη *μετα-μοντέρνα* περίοδο στην οποία η ΤΝ καλείται να παίξει ένα σημαντικό ρόλο σε ένα νέο πληροφοριακό περιβάλλον του οποίου κύρια χαρακτηριστικά είναι η εξάπλωση του διαδικτύου και η διείσδυση των υπολογιστικών συστημάτων σε κάθε είδους συσκευές ευρείας και καθημερινής χρήσης (pervasive computing).

Η Τεχνητή Νοημοσύνη Σήμερα (1/4)

❖ Τα τελευταία χρόνια είχαμε σημαντικές εξελίξεις σε εφαρμογές της Τ.Ν. Αυτήν τη στιγμή υπάρχουν:

- ❑ Συστήματα τα οποία βοηθούν τον χρήστη στο να χρησιμοποιήσει ορισμένα προγράμματα (για παράδειγμα Office Assistant),
- ❑ Να αναζητήσει πληροφορία στο διαδίκτυο, να στείλει email, να τηρήσει ραντεβού, να συγκρίνει τιμές προϊόντων και πολλά άλλα.
- ❑ Σε ορισμένες περιπτώσεις μάλιστα είναι σε θέση ακόμη και να μιλούν και να αστειεύονται (για παράδειγμα το BONZIBUDDY).
- ❑ Συστήματα αναγνώρισης φωνής (π.χ. Pegasus), τα οποία κλείνουν αεροπορικές θέσεις τηλεφωνικά, βρίσκοντας τις βέλτιστες πτήσεις ή δίνουν διάφορες πληροφορίες γενικού ενδιαφέροντος (π.χ. η φωνητική πύλη MyCosmos).
- ❑ Έμπειρα συστήματα πραγματικού χρόνου (π.χ. MARVEL) που επεξεργάζονται τα δεδομένα που μεταδίδονται από διαστημόπλοια.
- ❑ Ρομποτικά συστήματα που οδηγούν αυτοκίνητα σε αυτοκινητόδρομο χρησιμοποιώντας video κάμερες και sonar (σύστημα ALVIN)
- ❑ Συστήματα που διεξάγουν ιατρικές διαγνώσεις.
- ❑ Συστήματα που ελέγχουν και ρυθμίζουν την κυκλοφορία αυτοκινήτων.
- ❑ και πολλά άλλα.

Η Τεχνητή Νοημοσύνη Σήμερα (2/4)

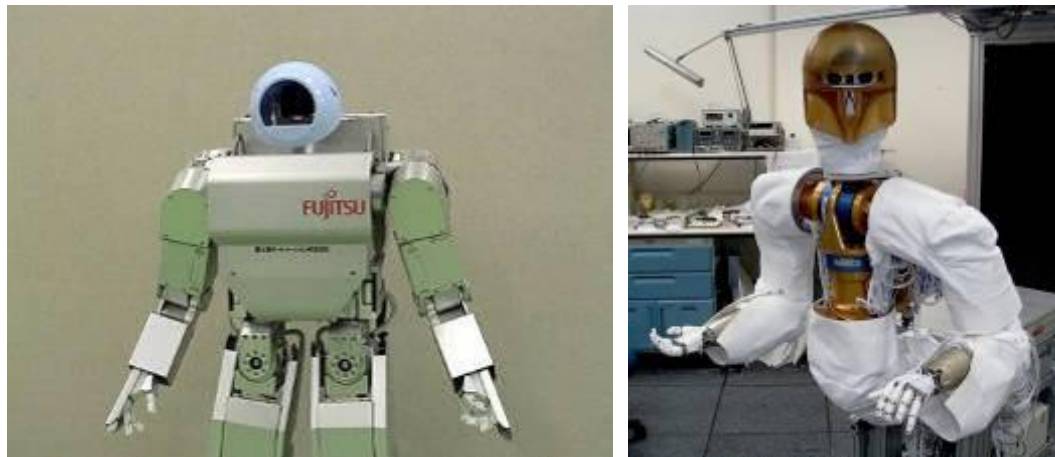
❖ Η εταιρεία SONY, ανέπτυξε το Biped Entertainment Robot - SDR -4X), το σκυλάκι AIBO με δυνατότητες αυτονομίας, αναγνώριση ομιλίας, έκφρασης συναισθημάτων με λόγο ή κινήσεις και το ρομπότ νέας γενιάς όπως το αποκαλεί, το QRIO το οποίο μπορεί και χορεύει και επικοινωνεί, αναγνωρίζοντας 10,000 ιαπωνικές λέξεις, αγγλικές ακόμη και ελληνικές.



Τα ρομπότ της SONY SDR-4X (αριστερά) και QRIO (δεξιά).

Η Τεχνητή Νοημοσύνη Σήμερα (3/4)

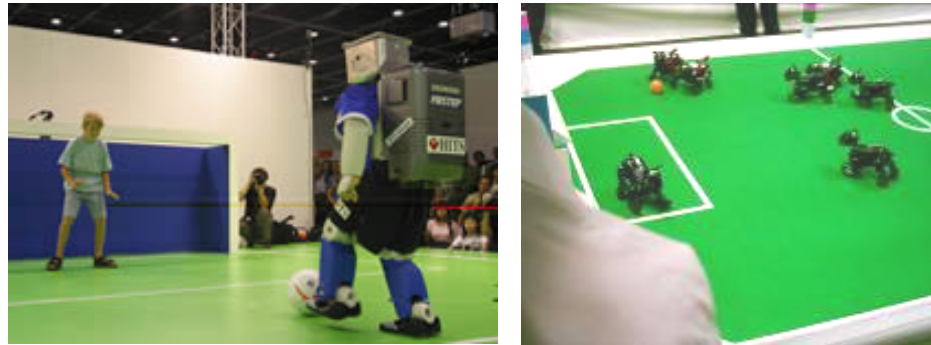
- ❖ Η εταιρεία FUJITSU ανέπτυξε το ανθρωποειδές ρομπότ HOAP (Humanoid for Open Architecture Platform) το οποίο μπορεί να κουνά το κεφάλι, τη μέση και τα χέρια του και μπορεί να συνδεθεί σε έναν υπολογιστή για μεταφορά δεδομένων.
- ❖ Η NASA σε συνεργασία με την Υπηρεσία Ανάπτυξης Προηγμένης Στρατιωτικής Τεχνολογίας των ΗΠΑ, (DARPA) ανέπτυξαν τον "Ρομποναύτη" (ROBONAUT) για τη συντήρηση του διαστημικού τηλεσκοπίου HUBBLE, προσαρμοσμένο πάνω στον ρομποτικό βραχίονα του διαστημικού λεωφορείου.



Τα ρομπότ HOAP (αριστερά) και ROBONAUT (δεξιά)

Η Τεχνητή Νοημοσύνη Σήμερα (4/4)

❖ Διοργανώνονται σε ετήσια βάση διεθνείς αγώνες ποδοσφαίρου, το ROBOCUP, στο οποίο πρωταγωνιστές είναι ρομπότ κάθε είδους (ανθρωποειδή, τετράποδα, κτλ.) με τελικό στόχο να αναπτυχθεί μια ομάδα αυτόνομων ανθρωποειδών ρομπότ μέχρι το 2050, ικανή να νικήσει στο ποδόσφαιρο την πρωταθλήτρια κόσμου.



Στιγμιότυπα από διοργανώσεις Robocup

Η ΤΝ στη λογοτεχνία και τον κινηματογράφο

- ❖ Η ΤΝ υπήρχε ανέκαθεν στο μυαλό των ανθρώπων.
 - Τα πρώτα δείγματά της εμφανίζονται στην ελληνική μυθολογία (Ο Ήφαιστος κατασκεύαζε ανθρωποειδή-υπηρέτες και ο Πυγμαλίωνας την ίδια του τη γυναίκα).
- ❖ Στη σύγχρονη εποχή, η ΤΝ απασχολεί τη λογοτεχνία και τον κινηματογράφο.
 - Ο κινηματογράφος αντιμετώπισε τα τεχνητά ευφυή όντα άλλοτε με φόβο (Terminator, I Robot, Matrix), άλλοτε με ελπίδα (Robocop, Bicentennial Man) και άλλοτε με συμπάθεια (Blade Runner, AI).
 - Δεν παρέλειψε βέβαια να εκφράσει και τις επιφυλάξεις του ως προς τη ηθική των ανθρώπων που κατασκευάζουν τέτοια όντα (Space Odyssey 2001, 2010).
- ❖ Η πραγματικότητα απέχει πολύ από την επιστημονική φαντασία.
 - Η κατασκευή υπολογιστών σαν τον HAL της ταινίας Space Odyssey 2001, ή ανδροειδών σαν τον David της ταινίας AI, αποτελεί, με τα σημερινά δεδομένα, ένα μακρινό όνειρο.

Επίλογος

- ❖ Η αλματώδης εξέλιξη των υπολογιστικών συστημάτων δημιουργεί συνεχώς νέες απαιτήσεις για τον τρόπο που αυτά πρέπει να επιλύουν προβλήματα.
- ❖ Πρωταρχικός στόχος ήταν η κατασκευή ενός συστήματος που θα μπορούσε να κερδίσει τον παγκόσμιο πρωταθλητή στο σκάκι.
- Επιτεύχθηκε το καλοκαίρι του 1997 με τη νίκη του DEEP BLUE επί του Kasparov.
- ❖ Η TN θέτει συνεχώς υψηλότερους στόχους και πλέον προσπαθεί να δημιουργήσει συστήματα που εξαρτώνται λιγότερο από τον προγραμματιστή και περισσότερο από την ικανότητά τους να μαθαίνουν πώς να συμπεριφέρονται, αλληλεπιδρώντας με το περιβάλλον.

Περιγραφή Προβλημάτων

❖ Διαισθητικά: υπάρχει μία δεδομένη κατάσταση (*αρχική*), υπάρχει μία επιθυμητή κατάσταση (*τελική*) και διαθέσιμες *ενέργειες* που πρέπει να γίνουν ώστε να φτάσουμε στην επιθυμητή.

Η επίλυση προβλημάτων που επιδιώκεται από την TN απαιτεί τον τυποποιημένο και σαφή ορισμό τους.

❖ Ο ορισμός ενός προβλήματος είναι ανεξάρτητη από την πολυπλοκότητα επίλυσής του η οποία αφορά τον αλγόριθμο αναζήτησης.

Κατηγορίες Προβλημάτων

❖ Πραγματικά και πολύπλοκα προβλήματα (*real world problems*):

- σκάκι (chess),
- πλανόδιος πωλητής (traveling salesperson),
- N-βασίλισσες (N-queens),
- σάκος (knapsack),

κλπ.

❖ Απλά προβλήματα (toy problems)

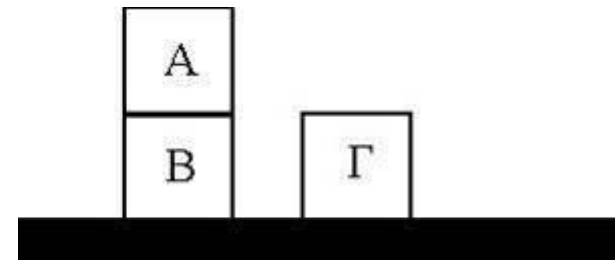
- κύβοι (blocks),
- Npuzzle,
- τρίλιζα (tic-tac-toe),
- λαβύρινθος (maze),
- πύργοι του Ανόι (Hanoi towers),
- κανίβαλοι και ιεραπόστολοι (missionaries and cannibals),
- ποτήρια (water glass)

κλπ.

❖ Η μεθοδολογία που εφαρμόζεται στην αναπαράσταση και την επίλυση είναι παρόμοια και στις δύο ακραίες περιπτώσεις προβλημάτων.

Παραδείγματα Απλών Προβλημάτων (1/2)

κύβοι (blocks)



Npuzzle

8	3	5
4	1	7
2		6

τρίλιζα (tic-tac-toe)

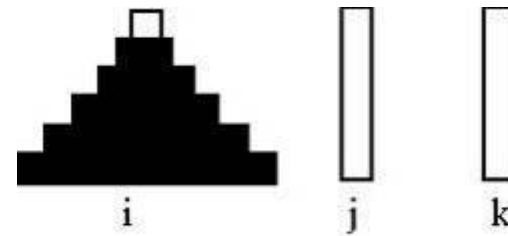
		o
o	x	
	x	

λαβύρινθος (maze)

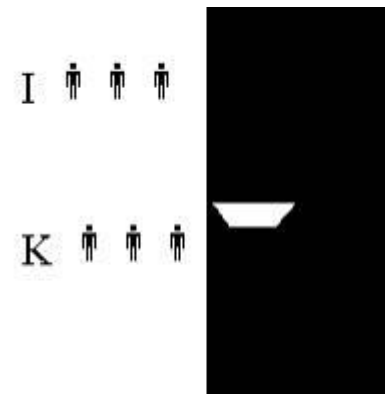


Παραδείγματα Απλών Προβλημάτων (2/2)

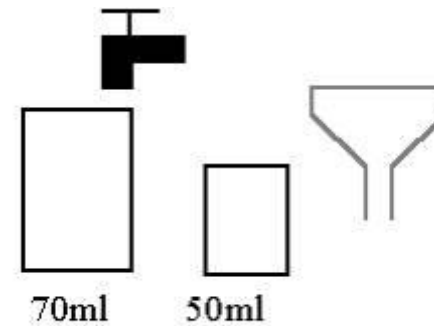
πύργοι του Ανόι (Hanoi towers)



κανίβαλοι και ιεραπόστολοι (missionaries and cannibals)



ποτήρια (water glass)



Περιγραφή Προβλημάτων

- ❖ Η περιγραφή ενός προβλήματος μπορεί να γίνει με δύο βασικούς τρόπους:
 - ❑ Περιγραφή με Χώρο Καταστάσεων (State Space) και
 - ❑ Περιγραφή με Αναγωγή (Reduction).

1.1.1

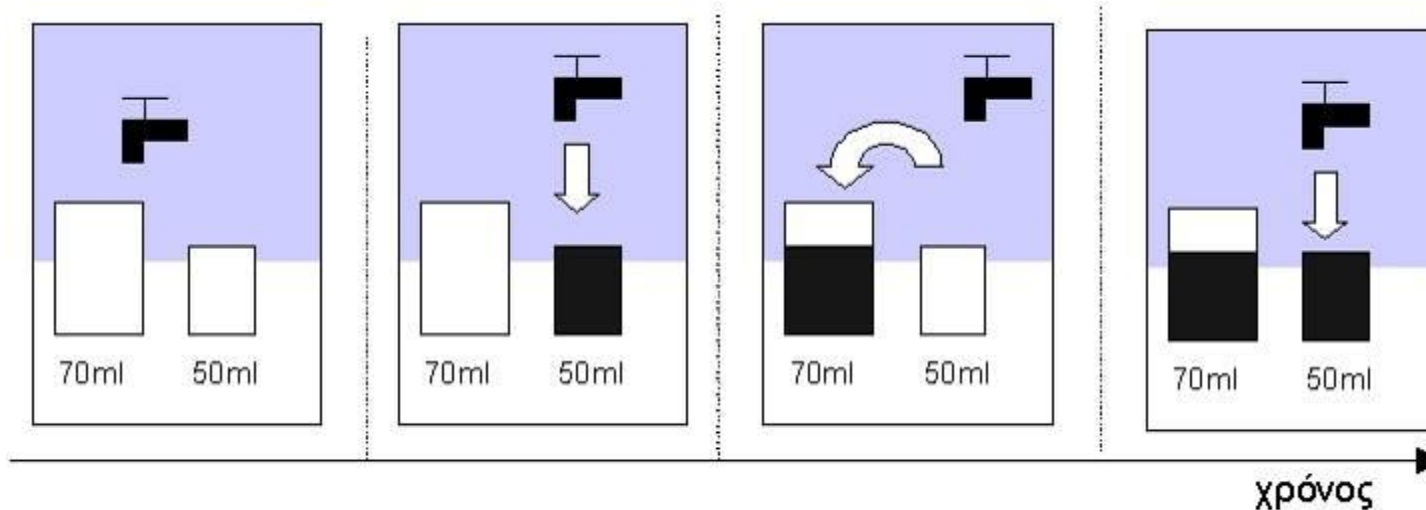
Περιγραφή Προβλημάτων με Χώρο Καταστάσεων (1/2)

❖ Ο κόσμος ενός προβλήματος αποτελείται από τα αντικείμενα, τις ιδιότητες των αντικειμένων και τις σχέσεις που τα συνδέουν

- ❑ Κλειστός κόσμος (closed world).
- ❑ Ανοιχτός κόσμος (open world) .

Κατάσταση προβλήματος

Κατάσταση ενός κόσμου είναι ένα στιγμιότυπο (*instance*) ή φωτογραφία (*snapshot*) μίας συγκεκριμένης χρονικής στιγμής της εξέλιξης του κόσμου.



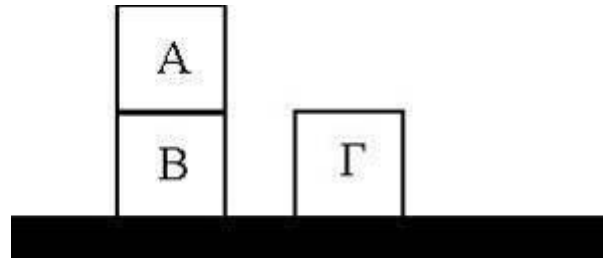
Περιγραφή Προβλημάτων με Χώρο Καταστάσεων (2/2)

Κατάσταση (state) ενός κόσμου είναι μία επαρκής αναπαράσταση του κόσμου σε μία δεδομένη χρονική στιγμή.

- "*Επαρκής*": ιδιότητες μιας κατάστασης πρέπει να επιλεγούν με τέτοιο τρόπο ώστε διαφορετικές τιμές των ιδιοτήτων να αντικατοπτρίζουν τις πραγματικές διαφορές των στιγμιότυπων του κόσμου (αφαίρεσης, abstraction).

Παράδειγμα

Ο κόσμος των κύβων



Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος A	Κύβος A είναι ελεύθερος	Κύβος A πάνω στον κύβο B
Κύβος B	Κύβος Γ είναι ελεύθερος	Κύβος B πάνω στο T
Κύβος Γ	T έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο T
T είναι Τραπέζι	Κύβος B δεν είναι ελεύθερος	

❖ Κόσμος του προβλήματος: Τρεις κύβοι και ένα τραπέζι.

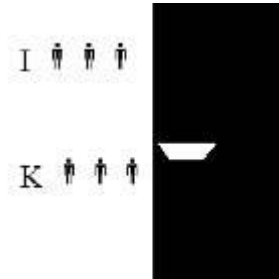
❖ Μια Κατάσταση:

Κύβος A πάνω στον κύβο B
Κύβος B πάνω στο T
Κύβος Γ πάνω στο T
Κύβος A ελεύθερος
Κύβος Γ ελεύθερος

❖ Άλλα χαρακτηριστικά (π.χ. χρώμα, βάρος, κλπ) είναι αδιάφορα.

Παράδειγμα

Ιεραπόστολοι και Κανίβαλοι



Αντικείμενα	Ιδιότητες	Σχέσεις
3 Ιεραπόστολοι 3 Κανίβαλοι Βάρκα Αριστερή Όχθη Δεξιά Όχθη	Βάρκα δύο ατόμων	Ιεραπόστολοι στην αριστερή όχθη Κανίβαλοι στην αριστερή όχθη Βάρκα στην αριστερή όχθη

Τελεστές Μετάβασης

- ❖ **Τελεστής μετάβασης** (*transition operator*) ή **ενέργεια** (*action*) είναι μια αντιστοίχιση μίας κατάστασης του κόσμου σε νέες καταστάσεις.
- ❖ **Παράδειγμα:** Στον κόσμο των κύβων, οι τελεστές μετάβασης είναι:
 - ❑ Βάλε τον κύβο A πάνω στον κύβο Γ.
 - ❑ Βάλε τον κύβο A πάνω στον κύβο B.
 - ❑ κλπ
- ❖ Στους τελεστές χρησιμοποιούμε και μεταβλητές.
 - ❑ **Παράδειγμα:** Βάλε κάποιον κύβο X πάνω σε κάποιον κύβο Y.
- ❖ Οι **Προϋποθέσεις** εφαρμογής (*preconditions*) που πρέπει να τηρούνται για να εφαρμοστεί ένας τελεστής.
- ❖ Η κατάσταση που προκύπτει πρέπει να είναι **Έγκυρη** (*valid*).

Τελεστές Μετάβασης

Παράδειγμα

Τελεστής:

Μετέφερε δύο ιεραπόστολους από την αριστερή όχθη στη δεξιά

Προϋποθέσεις:

Υπάρχουν τουλάχιστον 2 ιεραπόστολοι στην αριστερή όχθη.

Η βάρκα είναι στην αριστερή όχθη.

Ο αριθμός των ιεραποστόλων που θα προκύψει στην αριστερή όχθη να μην είναι μικρότερος από τον αριθμό των κανιβάλων ή να μην υπάρχει άλλος ιεραπόστολος στην αριστερή όχθη.

Αποτελέσματα:

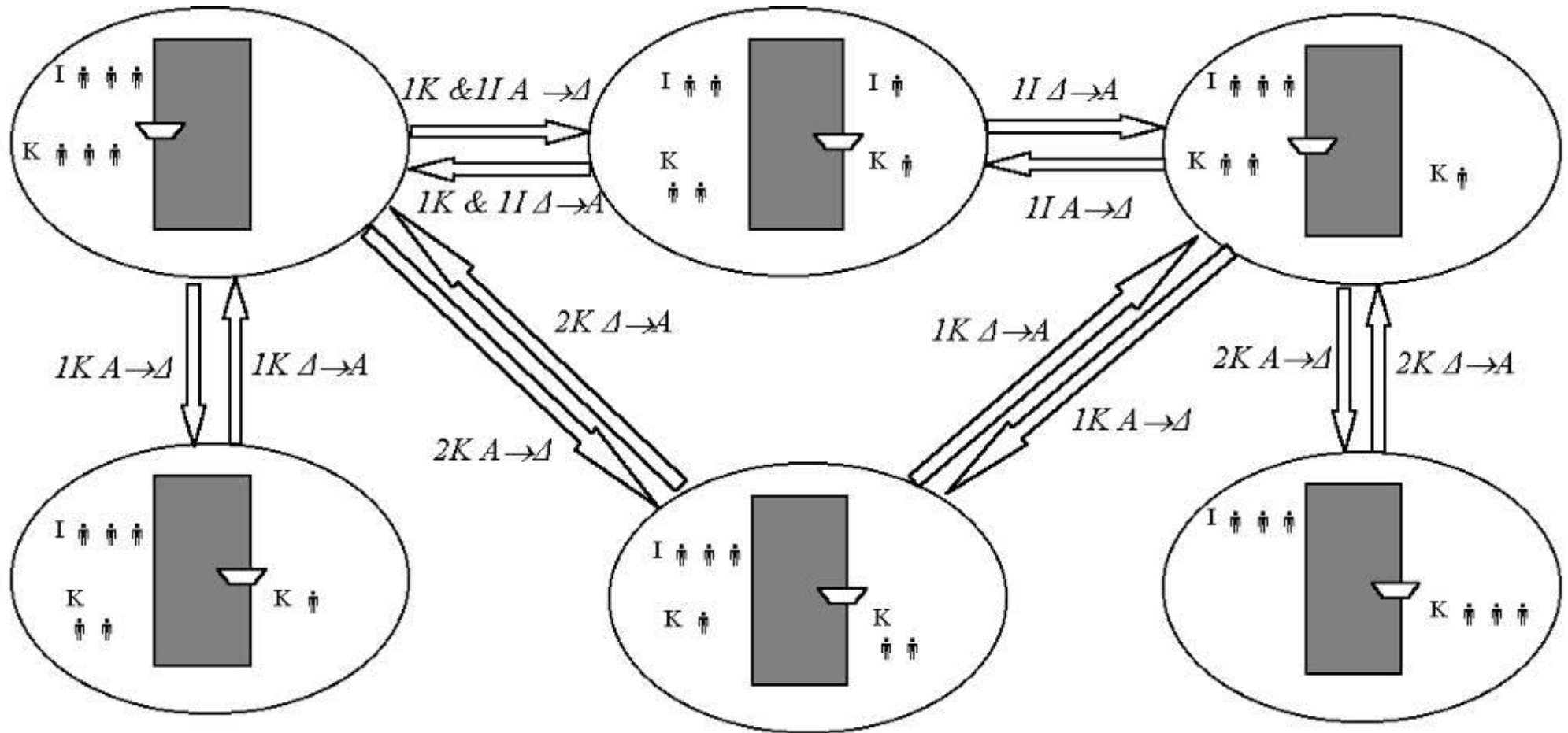
Ο αριθμός των ιεραποστόλων στην αριστερή όχθη μειώνεται κατά 2.

Ο αριθμός των ιεραποστόλων στην δεξιά όχθη αυξάνεται κατά 2.

Η βάρκα είναι πλέον δεξιά και όχι αριστερά

Χώρος Καταστάσεων

Χώρος καταστάσεων (*state space* ή *domain space*) ενός προβλήματος ονομάζεται το σύνολο όλων των έγκυρων καταστάσεων.



Αρχικές και Τελικές καταστάσεις

- ❖ Η αρχική (*initial state*) και τελική (*final* ή *goal state*) ή τελικές καταστάσεις εκφράζουν το δεδομένο και το ζητούμενο αντίστοιχα.

8	3	5
4	1	7
2		6

Αρχική Κατάσταση

1	2	3
4	5	6
7	8	

Τελική Κατάσταση

Ορισμός προβλήματος

- ❖ Ένα πρόβλημα (*Problem*) ορίζεται ως η τετράδα $P = (I, G, T, S)$ όπου:
 - ❑ I είναι η αρχική κατάσταση, $I \in S$
 - ❑ G είναι το σύνολο των τελικών καταστάσεων, $G \subseteq S$
 - ❑ T είναι το σύνολο των τελεστών μετάβασης, $T: S \leftrightarrow S$
 - ❑ S είναι ο χώρος καταστάσεων.

Λύση προβλήματος

Λύση (Solution) σε ένα πρόβλημα (I, G, T, S) , είναι μία ακολουθία από τελεστές μετάβασης $t_1, t_2, \dots, t_n \in T$ με την ιδιότητα $g = t_n(\dots(t_2(t_1(I))))$, όπου $g \in G$

❖ Λύση σε ένα πρόβλημα είναι η ακολουθία τελεστών που εφαρμόζονται στην αρχική κατάσταση για να προκύψει η τελική κατάσταση.

❖ **Παράδειγμα:**

Μετάφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη

Μετάφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη

Μετάφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη

Μετάφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετάφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη

Μετάφερε 1 ιεραπόστολο και 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετάφερε 2 ιεραπόστολους από την αριστερή στη δεξιά όχθη

Μετάφερε 1 κανίβαλο από τη δεξιά στην αριστερή όχθη

Μετάφερε 2 κανίβαλους από την αριστερή στη δεξιά όχθη

Μετάφερε 1 ιεραπόστολο από τη δεξιά στην αριστερή όχθη

Μετάφερε 1 ιεραπόστολο και 1 κανίβαλο από την αριστερή στη δεξιά όχθη

Κατηγορίες προβλημάτων (1/2)

- ❖ Κατηγοριοποίηση ανάλογα με την ερμηνεία του όρου "λύση".
- ❖ Προβλήματα όπου είναι πλήρως γνωστές οι τελικές καταστάσεις και επιδιώκεται η εύρεση μίας σειράς ενεργειών:
 - ❑ **προβλήματα σχεδιασμού ενεργειών** (*planning*) και προβλήματα πλοήγησης, στρατηγικής, εφοδιαστικής, κτλ.
- ❖ Προβλήματα όπου είναι γνωστές κάποιες ιδιότητες μόνο της τελικής κατάστασης και επιδιώκεται η εύρεση ενός πλήρους στιγμιότυπου της τελικής κατάστασης,
 - ❑ **προβλήματα χρονοπρογραμματισμού** (*scheduling*), σταυρόλεξα, κρυπτογραφικά, κτλ.
 - ❑ τα προβλήματα είναι γνωστά ως **προβλήματα ικανοποίησης περιορισμών** (*constraint satisfaction problems*).

Κατηγορίες προβλημάτων (2/2)

❖ Προβλήματα στα οποία είναι γνωστές κάποιες ιδιότητες μόνο της τελικής κατάστασης και επιδιώκεται η εύρεση μίας πλήρως γνωστής τελικής κατάστασης και η σειρά ενεργειών που θα οδηγήσουν σε αυτή:

□ **προβλήματα διαμόρφωσης** (*configuration*).

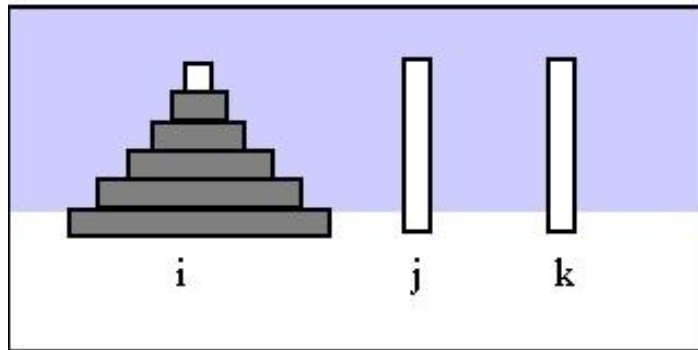
❖ Προβλήματα όπου είναι σχετικά εύκολο να βρεθούν λύσεις, αλλά το ζητούμενο είναι η βέλτιστη από αυτές.

□ **προβλήματα βελτιστοποίησης**, στα οποία και πάλι η τελική κατάσταση δεν είναι πλήρως γνωστή αλλά είναι γνωστά κάποια χαρακτηριστικά της.

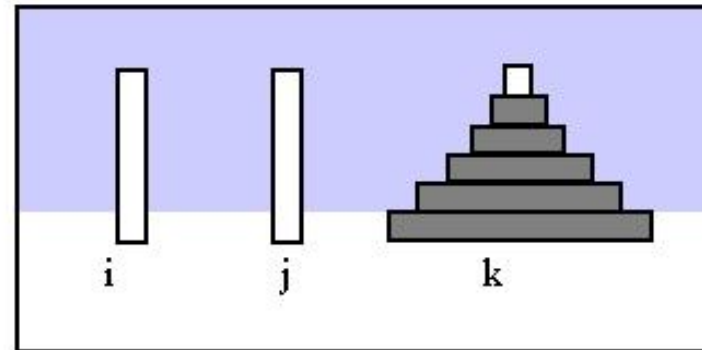
Περιγραφή με Αναγωγή (1/2)

- ❖ Μία ακολουθία από τελεστές ανάγουν την περιγραφή ενός προβλήματος σε υποπροβλήματα τα οποία είναι άμεσα επιλύσιμα, **αρχέγονα** (Primitive Problems).
- ❖ **Παράδειγμα:** Για να μεταφερθούν $n > 1$ δίσκοι από τον στύλο i στο στύλο k , πρέπει:
 - ❑ να μεταφερθούν $n-1$ δίσκοι από το i στο j ,
 - ❑ να μεταφερθεί 1 δίσκος από το i στο k ,
 - ❑ να μεταφερθούν $n-1$ δίσκοι από το j στο k .

Αρχική και τελική περιγραφή προβλήματος



Αρχική Περιγραφή



Τελική Περιγραφή

Περιγραφή με Αναγωγή (2/2)

- ❖ Αρχική Περιγραφή.
- ❖ Ένας Τελεστής Αναγωγής (reduction operator) ανάγει ένα πρόβλημα σε υποπροβλήματα.
- ❖ Τελική Περιγραφή.

Ορισμός προβλήματος

- ❖ Ένα πρόβλημα ορίζεται τυπικά ως η τετράδα $P = (ID, GD, TR, PP)$
 - ❑ όπου ID είναι η αρχική περιγραφή,
 - ❑ GD είναι ένα σύνολο από τελικές περιγραφές,
 - ❑ TR είναι ένα σύνολο τελεστών αναγωγής και
 - ❑ PP είναι ένα σύνολο από αρχέγονα προβλήματα.

1.2

Αλγόριθμοι Αναζήτησης

- ❖ Δοθέντος ενός προβλήματος με περιγραφή στο χώρο καταστάσεων ή με αναγωγή, στόχος είναι να βρεθεί η λύση του.
- ❖ Οι αλγόριθμοι που αναζητούν τη λύση σε ένα πρόβλημα ονομάζονται **αλγόριθμοι αναζήτησης** (*search algorithms*)
- ❖ Η επιλογή ενός αλγορίθμου αναζήτησης για ένα συγκεκριμένο πρόβλημα είναι σημαντική, διότι οι αλγόριθμοι αυτοί διαφέρουν μεταξύ τους σε αρκετά χαρακτηριστικά.

Αλγόριθμοι Αναζήτησης

Τυφλοί

Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα σε Πλάτος
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi-directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Branch and Bound	B&B	Επέκταση και Οριοθέτηση

1.2.1

Ευριστικοί

Hill Climbing	HC	Αναρρίχηση Λόφων
Best-First Search	BestFS	Αναζήτηση Πρώτα στο Καλύτερο
Beam Search	BS	Ακτινωτή Αναζήτηση
A* (A-star)	A*	A* (Άλφα Άστρο)

Παιχνιδιών 2 ατόμων

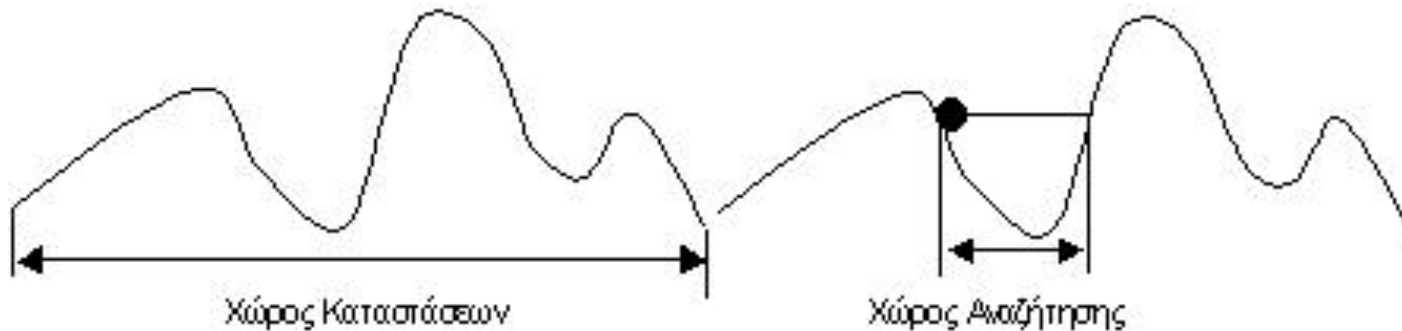
Minimax	Minimax	Αναζήτηση Μεγίστου-Ελαχίστου
Alpha-Beta	AB	Άλφα-Βήτα

Χώρος Αναζήτησης

Δοθέντος ενός προβλήματος (I, G, T, S) , *χώρος αναζήτησης* (*search space*) SP είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση.

Μία κατάσταση s ονομάζεται *προσβάσιμη* (*accessible*) αν υπάρχει μια ακολουθία τελεστών μετάβασης $t_1, t_2, \dots, t_k \in T$ τέτοια ώστε $s = t_k(\dots(t_2(t_1(I))))$.

- ❖ Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, δηλαδή $SP \subseteq S$.



- ❖ Ένας αλγόριθμος αναζήτησης δεν μειώνει τον χώρο αναζήτησης (που είναι δεδομένος) αλλά καθορίζει τον αριθμό των καταστάσεων που επισκέπτεται.

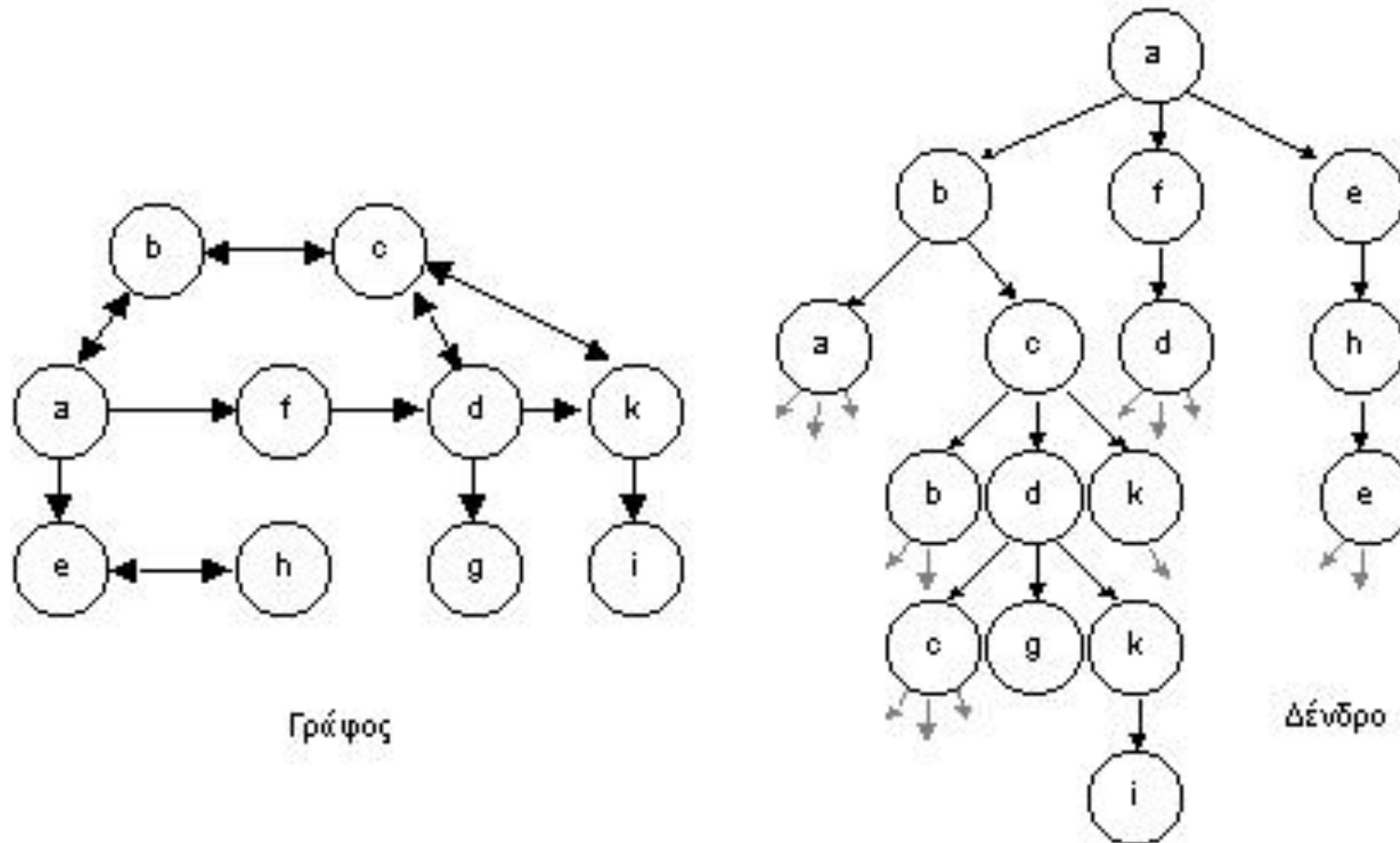
Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (1/2)

- ❖ Ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με γράφο.

Τμήμα Δένδρου	Αναπαράσταση
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Φύλλο (Tip, Leaf) ή Τερματικός κόμβος	Τελική Κατάσταση ή Αδιέξοδο (Dead Node), δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβασης.
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μια κατάσταση-Γονέα (Parent State) σε μία άλλη κατάσταση-Παιδί (Child State).
Λύση (Solution)	Μονοπάτι (Path) που ενώνει την αρχική με μία τελική κατάσταση
Επέκταση (Expansion)	Η διαδικασία παραγωγής όλων των καταστάσεων-παιδιών ενός κόμβου.
Παράγοντας Διακλάδωσης (Branching Factor)	Ο αριθμός των καταστάσεων-Παιδιών που προκύπτουν από την επέκταση μιας κατάστασης. Επειδή δεν είναι σταθερός αριθμός, αναφέρεται και ως Μέσος Παράγοντας Διακλάδωσης (Average Branching Factor).

Χώρος Αναζήτησης ως Δένδρο Αναζήτησης (2/2)

- ❖ Είναι πάντα εφικτό να μετατραπεί ο γράφος σε **δένδρο αναζήτησης** (*search tree*), το οποίο όμως μπορεί να έχει μονοπάτια απείρου μήκους.



Το φαινόμενο της εκθετικής αύξησης του αριθμού των κόμβων του δένδρου ονομάζεται συνδυαστική έκρηξη (*combinatorial explosion*).

Χαρακτηριστικά Αλγορίθμων (1/2)

- ❖ Ένας αλγόριθμος είναι μία αυστηρά καθορισμένη ακολουθία βημάτων-εντολών που επιδιώκει να λύσει ένα πρόβλημα.
- ❖ Δοθέντος ενός προβλήματος $P=(I,G,T,S)$ και μετά την εφαρμογή κάποιου αλγορίθμου στο χώρο αναζήτησής του, προκύπτει το **επιλυμένο πρόβλημα** (*solved problem*), το οποίο ορίζεται ως μία τετράδα $P_s=(V,A,F,G_s)$, όπου:
 - ❑ V είναι το σύνολο των καταστάσεων που εξέτασε ο αλγόριθμος αναζήτησης,
 - ❑ A είναι ο αλγόριθμος που χρησιμοποιήθηκε,
 - ❑ F είναι το σύνολο των λύσεων που βρέθηκαν, και
 - ❑ G_s είναι το σύνολο των τελικών καταστάσεων που εξετάστηκαν.
- ❖ Ο αριθμός των καταστάσεων που περιέχει το V και η σχέση του με το χώρο καταστάσεων S ενός προβλήματος και τον χώρο αναζήτησης SP , είναι ένα από τα χαρακτηριστικά της αποδοτικότητας του αλγορίθμου.

Χαρακτηριστικά Αλγορίθμων (2/2)

- ❖ Ένας αλγόριθμος ονομάζεται **εξαντλητικός** (*exhaustive*) όταν το σύνολο των καταστάσεων που εξετάζει ο αλγόριθμος για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης, δηλαδή $V=SP$.
- ❖ Ένας αλγόριθμος δεν λύνει πάντα κάποιο πρόβλημα, έστω και αν υπάρχει κάποια λύση. Τότε τα σύνολα G_s και F είναι κενά.
- ❖ Ένας αλγόριθμος αναζήτησης ονομάζεται **πλήρης** (*complete*) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν τέτοια λύση υπάρχει. Σε αντίθετη περίπτωση, ο αλγόριθμος ονομάζεται **μη-πλήρης** (*incomplete*).
- ❖ Μία λύση ονομάζεται **βέλτιστη** (*optimal*) αν οδηγεί στην καλύτερη, σύμφωνα με τη διάταξη, τελική κατάσταση. Όταν δεν υπάρχει διάταξη, μία λύση ονομάζεται βέλτιστη αν είναι η συντομότερη (*shortest*).
- ❖ Ένας αλγόριθμος αναζήτησης καλείται **αποδεκτός** (*admissible*) αν εγγυάται ότι θα βρει τη βέλτιστη λύση, αν μια τέτοια λύση υπάρχει.

1.2.3

Διαδικασία Επιλογής Αλγορίθμου Αναζήτησης

- ❖ Η επιλογή ενός αλγορίθμου βασίζεται στα εξής κριτήρια:
 - ❑ αριθμός των καταστάσεων που αυτός επισκέπτεται
 - ❑ δυνατότητα εύρεσης λύσεων εφόσον αυτές υπάρχουν
 - ❑ αριθμός των λύσεων
 - ❑ ποιότητα των λύσεων
 - ❑ αποδοτικότητά του σε χρόνο
 - ❑ αποδοτικότητά του σε χώρο (μνήμη)
 - ❑ ευκολία υλοποίησής του
 - ❑ κλάδεμα

Κλάδεμα ή αποκοπή καταστάσεων (pruning) του χώρου αναζήτησης είναι η διαδικασία κατά την οποία ο αλγόριθμος απορρίπτει, κάτω από ορισμένες συνθήκες, κάποιες καταστάσεις και μαζί με αυτές όλο το υποδένδρο που εκτυλίσσεται κάτω από τις καταστάσεις αυτές.

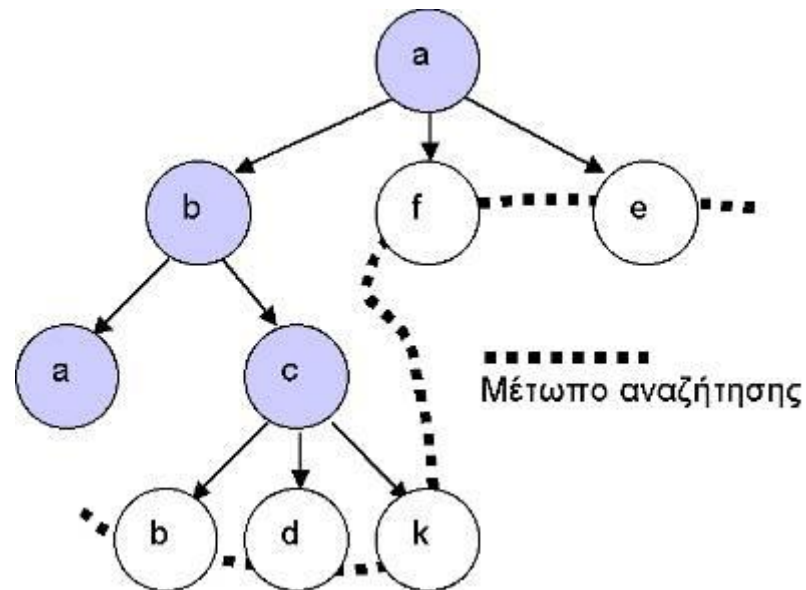
1.2.4

Γενικός Αλγόριθμος Αναζήτησης

❖ Οι αλγόριθμοι αναζήτησης χρησιμοποιούν κοινές δομές δεδομένων:

Μέτωπο της αναζήτησης (search frontier) ενός αλγορίθμου είναι η λίστα των καταστάσεων που ο αλγόριθμος έχει ήδη επισκεφτεί, αλλά δεν έχουν ακόμη επεκταθεί.

Κλειστό σύνολο (closed set) ενός αλγορίθμου αναζήτησης είναι το σύνολο όλων των καταστάσεων που έχουν ήδη επεκταθεί.



Γενικός Αλγόριθμος Αναζήτησης

- Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
- Αν το μέτωπο αναζήτησης είναι άδειο τότε σταμάτησε.
- Πάρε την πρώτη σε σειρά κατάσταση του μετώπου της αναζήτησης.
- Αν είναι η κατάσταση αυτή μέρος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
- Αν είναι η κατάσταση αυτή τελική κατάσταση τότε τύπωσε τη λύση και πήγαινε στο βήμα 2.
- Εφάρμοσε τους τελεστές μετάβασης για να παράγεις τις καταστάσεις-παιδιά.
- Βάλε τις νέες καταστάσεις-παιδιά στο μέτωπο της αναζήτησης.
- Κλάδεψε τις καταστάσεις που δε χρειάζονται (σύμφωνα με κάποιο κριτήριο), βγάζοντάς τες από το μέτωπο της αναζήτησης.
- Κάνε αναδιάταξη στο μέτωπο της αναζήτησης (σύμφωνα με κάποιο κριτήριο).
- Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
- Πήγαινε στο βήμα 2.

Γενικός Αλγόριθμος (ψευδοκώδικας)

```
algorithm general (InitialState, FinalState)
begin
  Closed  $\emptyset$ ;
  Frontier <InitialState>;
  CurrentStateFirst (Frontier);
  while CurrentState  $\notin$  FinalState do
    Frontierdelete (CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet then
      begin
        NextExpand (CurrentState);
        Frontierinsert (Next, Frontier);
        Frontierprune (Frontier);
        Frontierreorder (Frontier);
        ClosedClosed {CurrentState};
      end;
    if Frontier =  $\emptyset$  then return failure;
    CurrentStateFirst (Frontier);
  endwhile;
end.
```

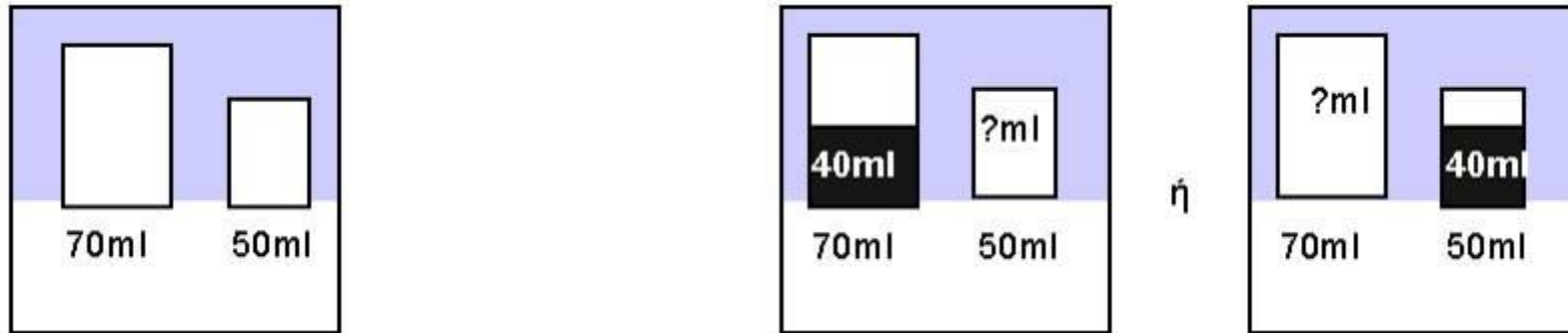
Αλγόριθμοι Τυφλής Αναζήτησης

- ❖ Οι αλγόριθμοι τυφλής αναζήτησης (*blind search algorithms*) εφαρμόζονται σε προβλήματα στα οποία δεν υπάρχει πληροφορία που να επιτρέπει αξιολόγηση των καταστάσεων.
- ❖ Στους αλγορίθμους τυφλής αναζήτησης έχει σημασία η χρονική σειρά με την οποία παράγονται οι καταστάσεις από το μηχανισμό επέκτασης.

Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα σε Πλάτος
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi-directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Branch and Bound	B&B	Επέκταση και Οριοθέτηση

Παράδειγμα

Το πρόβλημα των ποτηριών (1/2)



Τελεστής (1)
Γέμισε το ποτήρι των X ml μέχρι το χείλος από τη βρύση
Προϋποθέσεις
Το ποτήρι των X ml έχει 0 ml
Αποτελέσματα
Το ποτήρι των X ml έχει X ml

Παράδειγμα

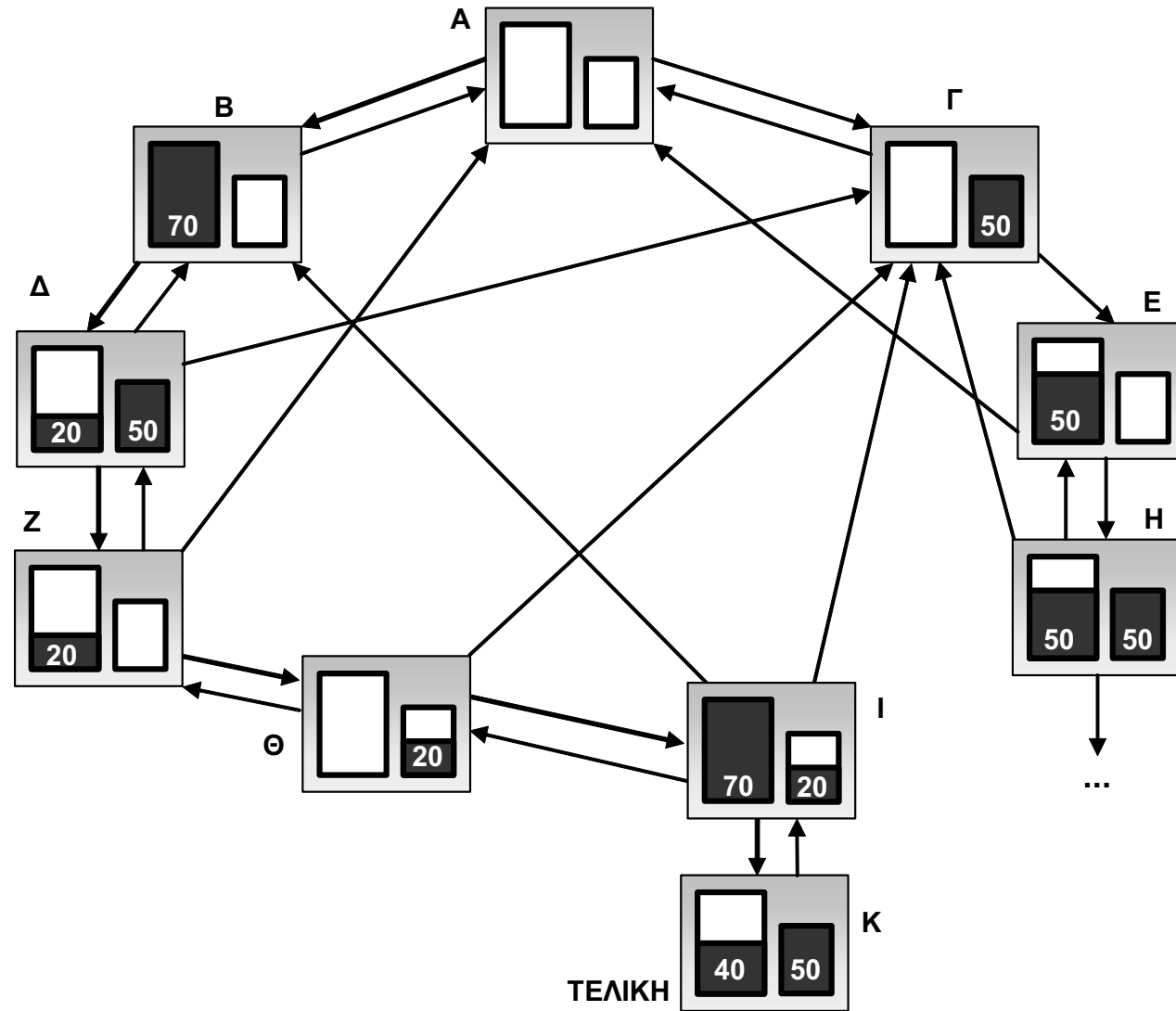
Το πρόβλημα των ποτηριών (2/2)

Τελεστής (2)
Γέμισε το ποτήρι των X ml από το ποτήρι των Y ml
Προϋποθέσεις
Το ποτήρι των X ml έχει Z ml Το ποτήρι των Y ml έχει W ml ($W \neq 0$)
Αποτελέσματα
Το ποτήρι των X ml έχει X ml και Το ποτήρι των Y ml έχει $W - (X - Z)$, αν $W \geq X - Z$ ή Το ποτήρι των X ml έχει $Z + W$ ml και Το ποτήρι των Y ml έχει 0, αν $W < X - Z$
Τελεστής (3)
Άδειασε το ποτήρι των X ml στο νεροχύτη
Προϋποθέσεις
Το ποτήρι έχει περιεχόμενο
Αποτελέσματα
Το ποτήρι των X ml έχει 0 ml

Μέρος του χώρου αναζήτησης

Στο πρόβλημα με τα ποτήρια

1.2.5



Αναζήτηση Πρώτα σε Βάθος

Ο αλγόριθμος πρώτα σε βάθος (*Depth-First Search - DFS*) επιλέγει προς επέκταση την κατάσταση που βρίσκεται πιο βαθιά στο δένδρο.

Ο αλγόριθμος DFS:

1. Βάλτε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλτε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
4. Αν η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία από τις τελικές, τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
8. Βάλτε τις καταστάσεις-παιδιά στην αρχή του μετώπου της αναζήτησης.
9. Βάλτε την κατάσταση-γονέα στο κλειστό σύνολο.

10. Πήγαινε στο βήμα 2.

Ο αλγόριθμος DFS (Ψευδοκώδικας)

```
algorithm dfs(InitialState, FinalStates)
begin
  Closed $\emptyset$ ;
  Frontier<InitialState>;
  CurrentStateFirst(Frontier);
  while CurrentState  $\notin$  FinalStates do
    Frontierdelete(CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet then
      begin
        ChildrenStates Expand(CurrentState);
        FrontierChildrenStates  $\wedge$  Frontier;
        ClosedClosed{CurrentState};
      end;
      if Frontier= $\emptyset$  then exit;
      CurrentStateFirst(Frontier);
    endwhile;
  return success;
end.
```

Αναζήτηση Πρώτα σε Βάθος

Σχόλια

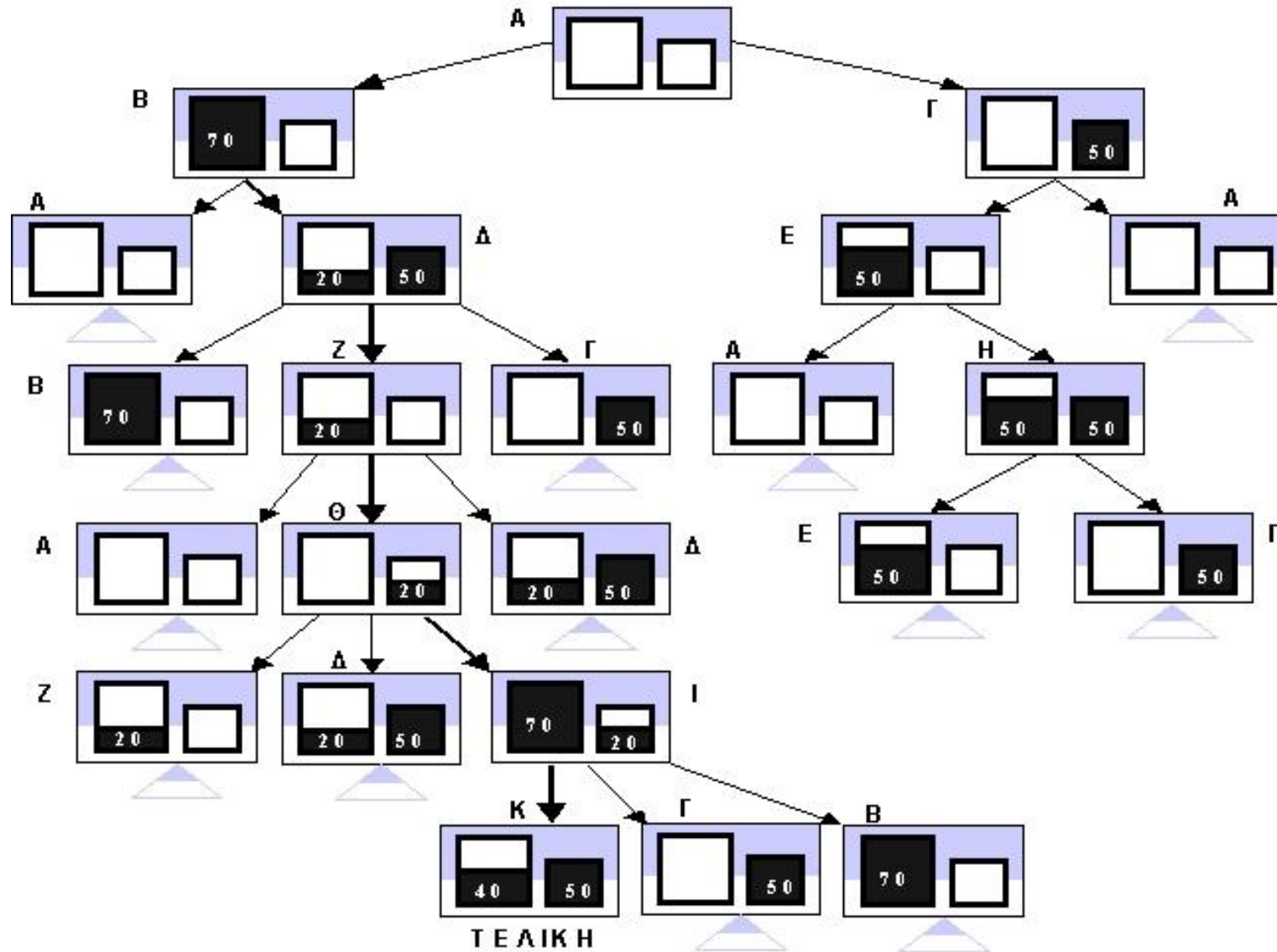
- ❖ Το μέτωπο της αναζήτησης είναι μια δομή στοίβας (Stack LIFO)
- ❖ Η εξέταση αμέσως προηγούμενων (χρονικά) καταστάσεων ονομάζεται χρονική οπισθοδρόμηση (*chronological backtracking*).

- ❖ Πλεονεκτήματα:
 - ❑ το μέτωπο της αναζήτησης δε μεγαλώνει πάρα πολύ.

- ❖ Μειονεκτήματα:
 - ❑ Δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη.
 - ❑ Θεωρείται μη-πλήρης.
 - ❑ Όταν όμως ο χώρος αναζήτησης είναι πεπερασμένος και χρησιμοποιείται κλειστό σύνολο, ο DFS θα βρει λύση, εάν μια τέτοια υπάρχει.

Αναζήτηση Πρώτα σε Βάθος (DFS)

Δένδρο αναζήτησης στο πρόβλημα των ποτηριών



Αναζήτηση Πρώτα σε Βάθος (DFS)

Πρόβλημα των ποτηριών

Μέτωπο της αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά
<A>	{}	A	<B, Γ>
<B, Γ>	{A}	B	<A, Δ>
<A, Δ, Γ>	{A,B}	A	- (βρόχος)
<Δ, Γ>	{A,B}	Δ	<B,Z,Γ>
<B,Z,Γ,Γ>	{A,B,Δ}	B	- (βρόχος)
<Z,Γ,Γ>	{A,B,Δ}	Z	<A,Θ,Δ>
<A,Θ,Δ,Γ,Γ>	{A,B,Δ,Z}	A	- (βρόχος)
<Θ,Δ,Γ,Γ>	{A,B,Δ,Z}	Θ	<Z,Δ,I>
<Z,Δ,I,Δ,Γ,Γ>	{A,B,Δ,Z,Θ}	Z	- (βρόχος)
<Δ,I,Δ,Γ,Γ>	{A,B,Δ,Z,Θ}	Δ	- (βρόχος)
<I,Δ,Γ,Γ>	{A,B,Δ,Z,Θ}	I	<K,Γ,B>
<K,Γ,B,Δ,Γ,Γ>	{A,B,Δ,Z,Θ,I}	K	ΤΕΛΙΚΗ

1.2.6

Αναζήτηση Πρώτα σε Πλάτος

Ο αλγόριθμος αναζήτησης πρώτα σε πλάτος (*Breadth First Search - BFS*) εξετάζει πρώτα όλες τις καταστάσεις που βρίσκονται στο ίδιο βάθος και μετά συνεχίζει στην επέκταση καταστάσεων στο αμέσως επόμενο επίπεδο.

Ο αλγόριθμος BFS:

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
4. Αν είναι η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μεταφοράς για να βρεις τις καταστάσεις-παιδιά.

8. Βάλε τις καταστάσεις-παιδιά στο τέλος του μετώπου της αναζήτησης.

9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.

10. Πήγαινε στο βήμα 2.

Ο αλγόριθμος BFS (Ψευδοκώδικας)

```
algorithm bfs(InitialState, FinalStates)
begin
  Closed $\emptyset$ ;
  Frontier<InitialState>;
  CurrentStateFirst(Frontier);
  while CurrentState  $\notin$  FinalStates do
    Frontierdelete(CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet
      begin
        ChildrenStates Expand(CurrentState);
        Frontier Frontier  $\cup$  ChildrenStates;
        ClosedClosed{CurrentState};
      end;
    if Frontier=  $\emptyset$  then exit;
    CurrentStateFirst(Frontier);
  endwhile;
  return success;
end.
```

Αναζήτηση Πρώτα σε Πλάτος

Σχόλια

- ❖ Το μέτωπο της αναζήτησης είναι μια δομή ουράς (Queue FIFO).
- ❖ Πλεονεκτήματα:
 - ❑ Βρίσκει πάντα την καλύτερη λύση (μικρότερη σε μήκος).
 - ❑ Είναι πλήρης.
- ❖ Μειονεκτήματα:
 - ❑ Το μέτωπο της αναζήτησης μεγαλώνει πολύ σε μέγεθος.

Αναζήτηση Πρώτα σε Πλάτος (BFS)

Πρόβλημα των ποτηριών (1/2)

Μέτωπο αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά
<A>	{}	A	<B, Γ>
<B, Γ>	{A}	B	<A, Δ>
<Γ, A, Δ>	{A, B}	Γ	<E, A>
<A, Δ, E, A>	{A, B, Γ}	A	- (βρόχος)
<Δ, E, A>	{A, B, Γ}	Δ	<B, Z, Γ>
<E, A, B, Z, Γ>	{A, B, Γ, Δ}	E	<A, H>
<A, B, Z, Γ, A, H>	{A, B, Γ, Δ, E}	A	- (βρόχος)
<B, Z, Γ, A, H>	{A, B, Γ, Δ, E}	B	- (βρόχος)
<Z, Γ, A, H>	{A, B, Γ, Δ, E}	Z	<A, Θ, Δ>
<Γ, A, H, A, Θ, Δ>	{A, B, Γ, Δ, E, Z}	Γ	- (βρόχος)
<A, H, A, Θ, Δ>	{A, B, Γ, Δ, E, Z}	A	- (βρόχος)
<H, A, Θ, Δ>	{A, B, Γ, Δ, E, Z}	H	<E, Γ>
<A, Θ, Δ, E, Γ>	{A, B, Γ, Δ, E, Z, H}	A	- (βρόχος)
<Θ, Δ, E, Γ>	{A, B, Γ, Δ, E, Z, H}	Θ	<Z, Δ, I>

Αναζήτηση Πρώτα σε Πλάτος (BFS)

Πρόβλημα των ποτηριών (2/2)

Μέτωπο αναζήτησης	Κλειστό Σύνολο	Κατάσταση	Παιδιά
<Δ,Ε,Γ,Ζ,Δ,Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Δ	- (βρόχος)
<Ε,Γ,Ζ,Δ,Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Ε	- (βρόχος)
<Γ,Ζ,Δ,Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Γ	- (βρόχος)
<Ζ,Δ,Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Ζ	- (βρόχος)
<Δ,Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Δ	- (βρόχος)
<Ι>	{Α,Β,Γ,Δ,Ε,Ζ,Η}	Ι	<Κ,Γ,Β>
<Κ,Γ,Β>	{Α,Β,Γ,Δ,Ε,Ζ,Η,Ι}	Κ	ΤΕΛΙΚΗ

Αλγόριθμος Επαναληπτικής Εκβάθυνσης

Ο αλγόριθμος επαναληπτικής εκβάθυνσης (Iterative Deepening - ID) συνδυάζει με τον καλύτερο τρόπο τους DFS και BFS.

Ο αλγόριθμος ID:

1. Όρισε το αρχικό βάθος αναζήτησης (συνήθως 1).
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.

Ο αλγόριθμος ID (Ψευδοκώδικας)

```
algorithm id(InitialState, FinalStates)
begin
  depth1
  while solution is not found do
    bounded_dfs(InitialState, FinalStates, depth) ;
    depthdepth+1
  endwhile ;
end.
```

Αναζήτηση ID

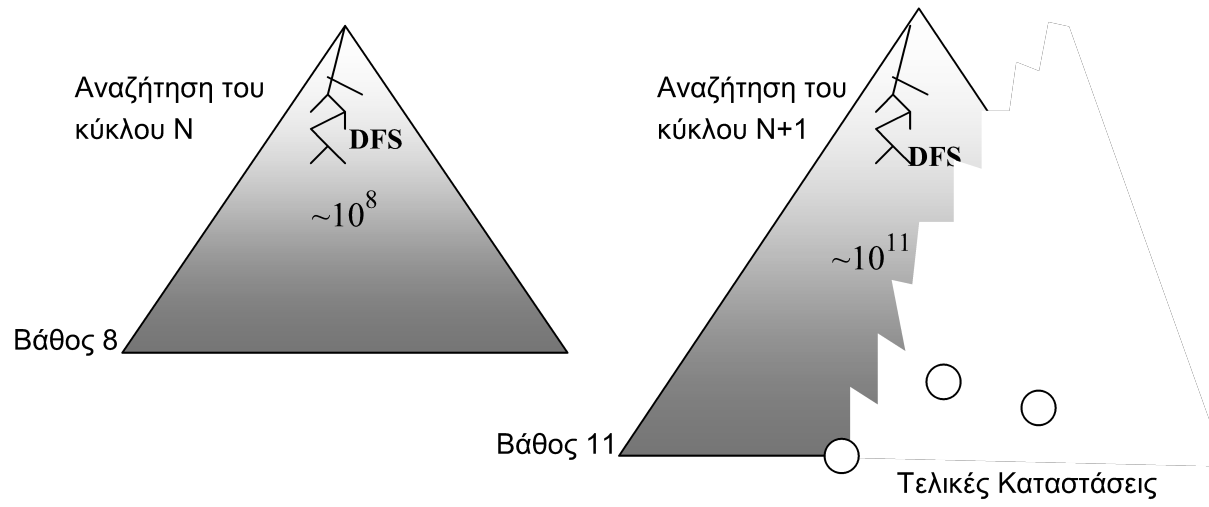
Σχόλια

❖ Μειονεκτήματα:

- Δε θυμάται τίποτα από τον προηγούμενο κύκλο αναζήτησης.

❖ Πλεονεκτήματα:

- Είναι πλήρης.
- Αν το βάθος αυξάνεται κατά 1 σε κάθε κύκλο και ο ID βρει λύση, τότε αυτή η λύση θα είναι η καλύτερη.
- Έχει αποδειχθεί ότι έχει την ίδια πολυπλοκότητα σε χώρο και χρόνο με τους DFS και BFS.
- Δεν κινδυνεύει να χαθεί σε κάποιο κλαδί απείρου μήκους.



Αναζήτηση Διπλής Κατεύθυνσης (1/2)

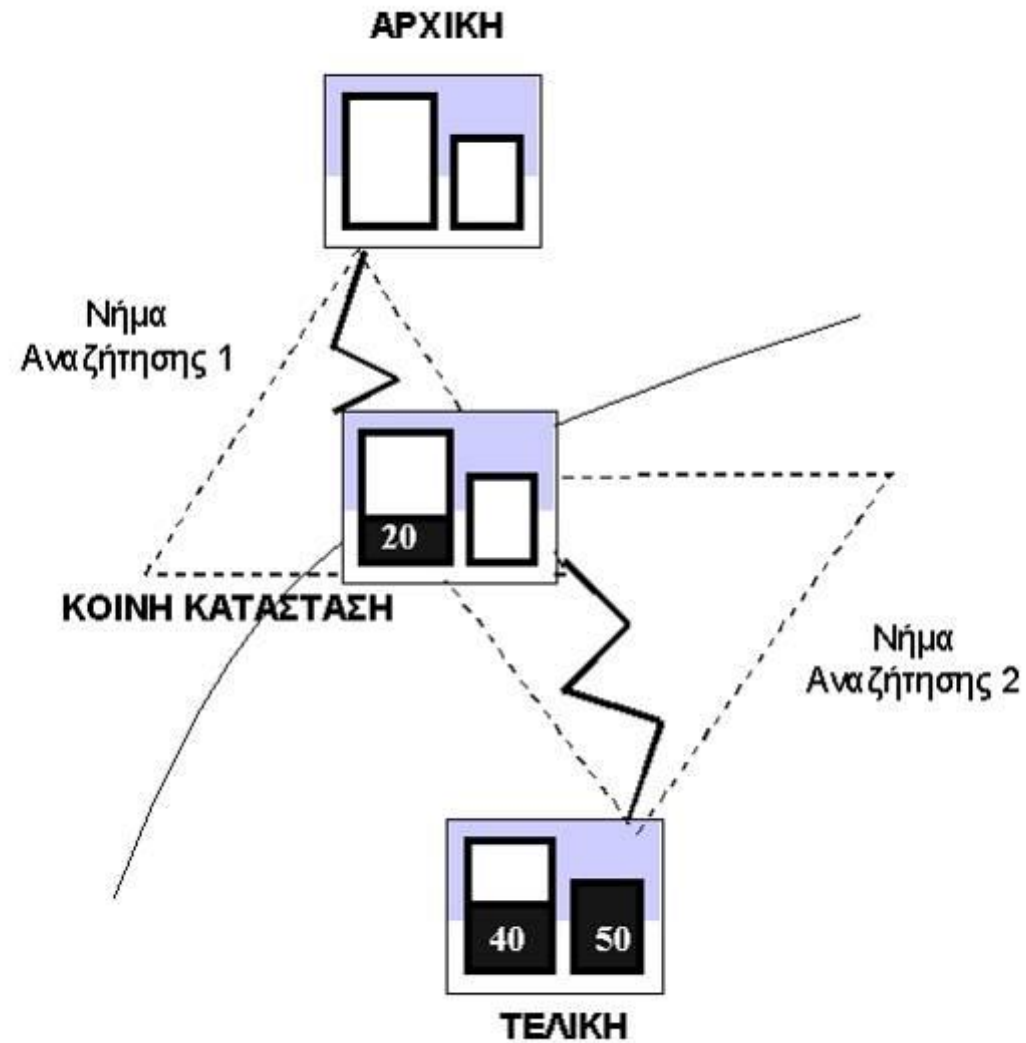
Η ιδέα της αναζήτησης διπλής κατεύθυνσης (Bidirectional Search - *BiS*) πηγάζει από τη δυνατότητα του παραλληλισμού (parallelism) στα υπολογιστικά συστήματα.

- ❖ Προϋποθέσεις κάτω από τις οποίες μπορεί να εφαρμοστεί:
 - ❑ Οι τελεστές μετάβασης είναι αντιστρέψιμοι (reversible), και
 - ❑ Είναι πλήρως γνωστή η τελική κατάσταση.

- ❖ Εφαρμογή
 - ❑ Αρχίζει την αναζήτηση από την αρχική και τελική κατάσταση ταυτόχρονα.
 - ❑ Αν κάποια κατάσταση που επεκτείνεται είναι κοινή, τότε βρέθηκε λύση.
 - ❑ Λύση είναι η ένωση των μονοπατιών από την κοινή κατάσταση έως την αρχική και έως την τελική κατάσταση.

- ❖ Μειονεκτήματα:
 - ❑ Υπάρχει επιπλέον κόστος επικοινωνίας μεταξύ των δύο αναζητήσεων.

Αναζήτηση Διπλής Κατεύθυνσης (2/2)



Επέκταση και Οριοθέτηση (B&B)

Ο αλγόριθμος επέκτασης και οριοθέτησης (Branch and Bound - *B&B*) εφαρμόζεται σε προβλήματα όπου αναζητείται η βέλτιστη λύση (ελάχιστο κόστος).

- ❖ Ο B&B κλαδεύει καταστάσεις (pruning) και μειώνει το χώρο αναζήτησης.
- ❖ Παράδειγμα: βρέθηκε μια λύση με κόστος 159. Κατά την αναζήτηση για άλλες λύσεις μια κατάσταση έχει ήδη κόστος 167. Άρα κλαδεύεται αυτή η κατάσταση.

Ο αλγόριθμος B&B:

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αρχική τιμή της καλύτερης λύσης είναι το + (όριο).
3. Αν το μέτωπο της αναζήτησης είναι κενό, τότε σταμάτησε.
Η καλύτερη μέχρι τώρα λύση είναι και η βέλτιστη.
4. Βγάλε την πρώτη σε σειρά κατάσταση από το μέτωπο της αναζήτησης.
5. Αν η κατάσταση ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 3.
6. Αν η κατάσταση είναι τελική, τότε ανανέωσε τη λύση ως την καλύτερη μέχρι τώρα και ανανέωσε την τιμή του ορίου με την τιμή που αντιστοιχεί στην τελική κατάσταση. Πήγαινε στο 3.
7. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά και την τιμή που αντιστοιχεί σε αυτές.
8. Βάλε τις καταστάσεις-παιδιά, των οποίων η τιμή δεν υπερβαίνει το όριο, μπροστά στο μέτωπο της αναζήτησης. (*)

9. Βάλτε την κατάσταση-γονέα στο κλειστό σύνολο.

10. Πήγαινε στο 3.

(*) Αυτός είναι DFS-B&B γιατί οι νέες καταστάσεις μπαίνουν μπροστά στο μέτωπο αναζήτησης. Υπάρχει και BFS-B&B.

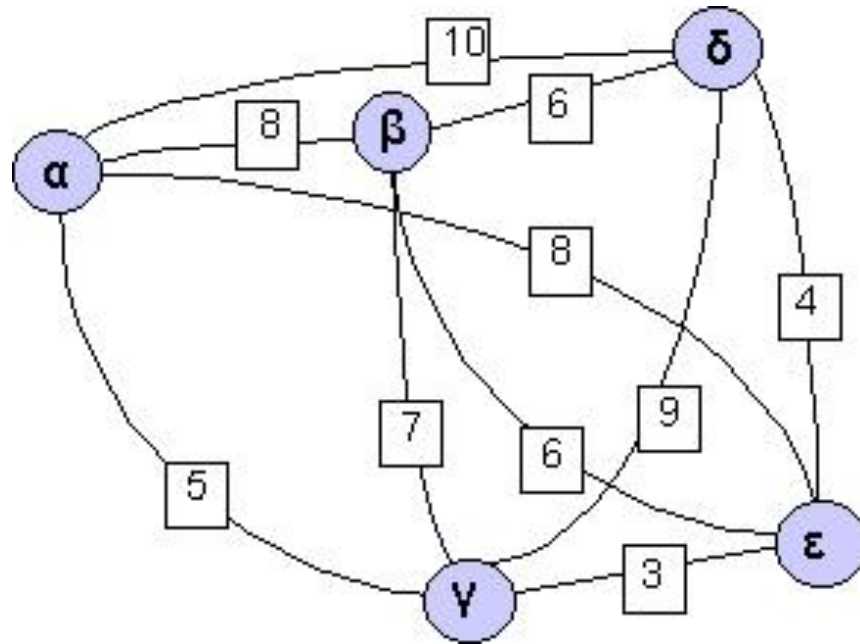
Ο αλγόριθμος B&B (Ψευδοκώδικας)

```
algorithm b&b(InitialState, FinalStates)
begin
  Closed $\emptyset$ ;
  Frontier<InitialState>;
  BestCost;
  BestState null;
  while Frontier  $\neq \emptyset$  do
    CurrentStateFirst(Frontier);
    CurrentCostCost(Current_State);

    Frontierdelete(CurrentState,Frontier);
    if CurrentCost < BestCost then
      if CurrentState  $\in$  FinalStates then
        BestStateCurrentState;
        BestCostCurrentCost;
      else
```

```
        NextExpand(CurrentState);
        ChildrenStates
        {s| s $\in$ Next  $\wedge$  s $\notin$ Frontier  $\wedge$  s $\notin$ Closed};
        FrontierChildrenStates ^
  Frontier;
    ClosedClosed{CurrentState};
    endif;
  endif;
endwhile;
  if BestState = null
    then return fail
    else return BestState and BestCost;
end.
```

Το Πρόβλημα του Πλανόδιου Πωλητή (TSP)



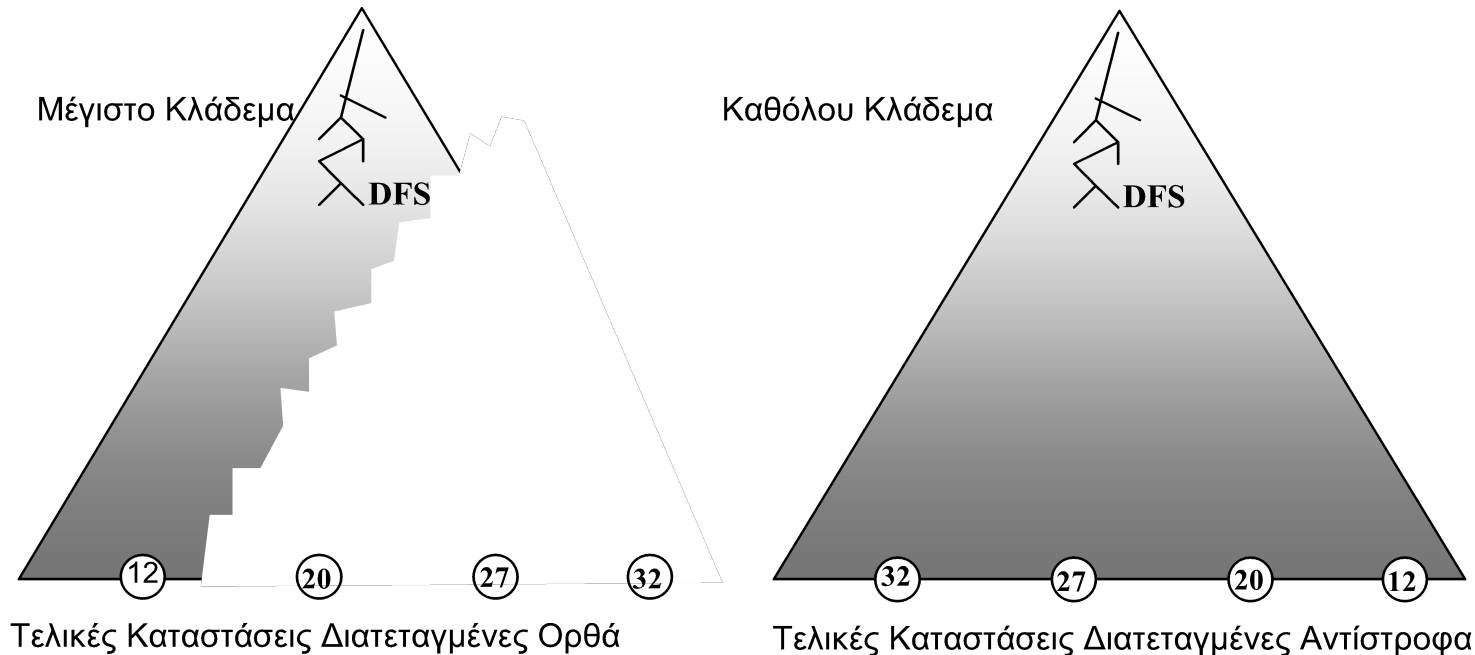
- ❖ Το πρόβλημα ανήκει στην κατηγορία *μη-πολυωνυμικού χρόνου (NP-complete)*.
 - the most notable characteristic of NP-complete problems is that no fast solution to them is known
 - the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows
 - the time required to solve even moderately large versions of many of these problems easily reaches into the billions or trillions of years, using any amount of computing power available today
 - determining whether or not it is possible to solve these problems quickly, called the P versus NP problem, is one of the principal unsolved problems in computer science today
- ❖ Το πρόβλημα είναι πρόβλημα ελαχιστοποίησης κόστους και έχει πολλές εφαρμογές.

Ο αλγόριθμος B&B στο πρόβλημα TSP

Μέτωπο της αναζήτησης	Κόστος Λύσης	Κατάσταση	Παιδιά
$\langle \alpha \rangle$	+	α	$\alpha\beta^8, \alpha\gamma^5, \alpha\delta^{10}, \alpha\epsilon^8$
$\langle \alpha\beta^8, \alpha\gamma^5, \alpha\delta^{10}, \alpha\epsilon^8 \rangle$	+	$\alpha\beta$	$\alpha\beta\gamma^{15}, \alpha\beta\delta^{14}, \alpha\beta\epsilon^{14}$
$\langle \alpha\beta\gamma^{15}, \alpha\beta\delta^{14}, \alpha\beta\epsilon^{14}, \alpha\gamma^5, \dots \rangle$	+	$\alpha\beta\gamma$	$\alpha\beta\gamma\delta^{24}, \alpha\beta\gamma\epsilon^{18}$
$\langle \alpha\beta\gamma\delta^{24}, \alpha\beta\gamma\epsilon^{18}, \alpha\beta\delta^{14}, \alpha\beta\epsilon^{14}, \dots \rangle$	+	$\alpha\beta\gamma\delta$	$\alpha\beta\gamma\delta\epsilon^{28}$
$\langle \alpha\beta\gamma\delta\epsilon^{28}, \alpha\beta\gamma\epsilon^{18}, \alpha\beta\delta^{14}, \dots \rangle$	+	$\alpha\beta\gamma\delta\epsilon$	$\alpha\beta\gamma\delta\epsilon\alpha^{36}$
$\langle \alpha\beta\gamma\delta\epsilon\alpha^{36}, \alpha\beta\gamma\epsilon^{18}, \alpha\beta\delta^{14}, \dots \rangle$	36	$\alpha\beta\gamma\delta\epsilon\alpha$	Τελική Κατάσταση
$\langle \alpha\beta\gamma\epsilon^{18}, \alpha\beta\delta^{14}, \dots \rangle$	36	$\alpha\beta\gamma\epsilon$	$\alpha\beta\gamma\epsilon\delta^{22}$
$\langle \alpha\beta\gamma\epsilon\delta^{22}, \alpha\beta\delta^{14}, \dots \rangle$	36	$\alpha\beta\gamma\epsilon\delta$	$\alpha\beta\gamma\epsilon\delta\alpha^{32}$
$\langle \alpha\beta\gamma\epsilon\delta\alpha^{32}, \alpha\beta\delta^{14}, \alpha\beta\epsilon^{14}, \dots \rangle$	32	$\alpha\beta\gamma\epsilon\delta\alpha^{32}$	Τελική Κατάσταση
...
$\langle \alpha\beta\delta\epsilon\gamma\alpha^{26}, \dots \rangle$	26	$\alpha\beta\delta\epsilon\gamma\alpha$	Τελική Κατάσταση
...
$\langle \alpha\beta\epsilon\gamma\delta^{26}, \dots \rangle$	26	$\alpha\beta\epsilon\gamma\delta$	Κλάδεμα
....
$\langle \alpha\epsilon\beta\gamma\delta^{30}, \dots \rangle$	26	$\alpha\epsilon\beta\gamma\delta$	Κλάδεμα
...
$\langle \rangle$	Ελάχιστη Τιμή	ΤΕΛΟΣ	

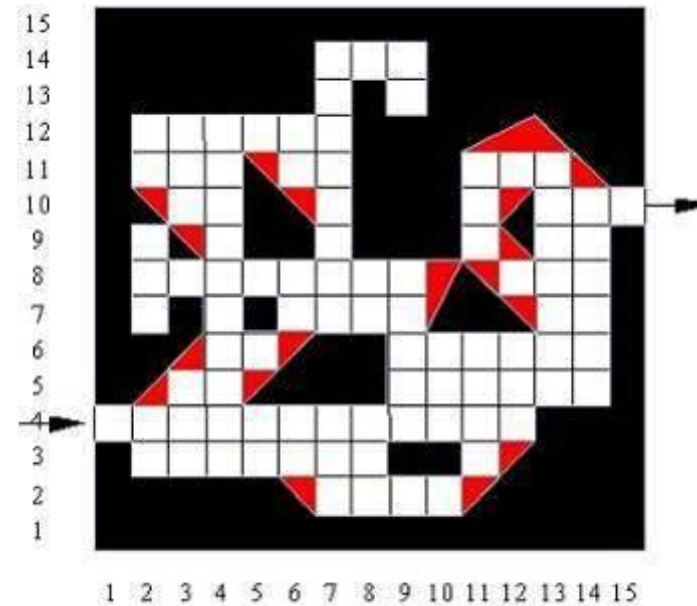
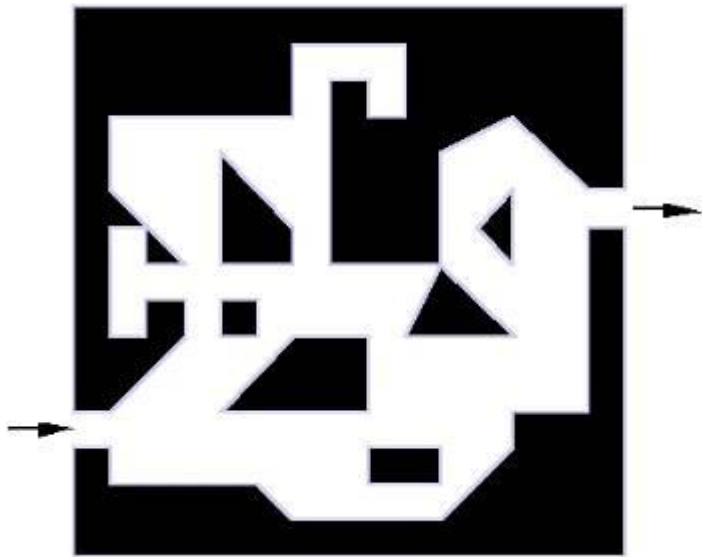
Ο αλγόριθμος B&B

- ❖ Ο B&B εφαρμόζεται όταν υπάρχει μια πραγματική εκτίμηση του κόστους.
- ❖ Το κέρδος από το κλάδεμα εξαρτάται από το πόσο γρήγορα θα βρεθεί μια καλή λύση.
- ❖ Υπάρχει περίπτωση να μη γίνει καθόλου κλάδεμα αν οι λύσεις είναι διατεταγμένες από τη χειρότερη προς την καλύτερη.
- ❖ Στη χειρότερη περίπτωση συμπεριφέρεται σαν τον DFS.



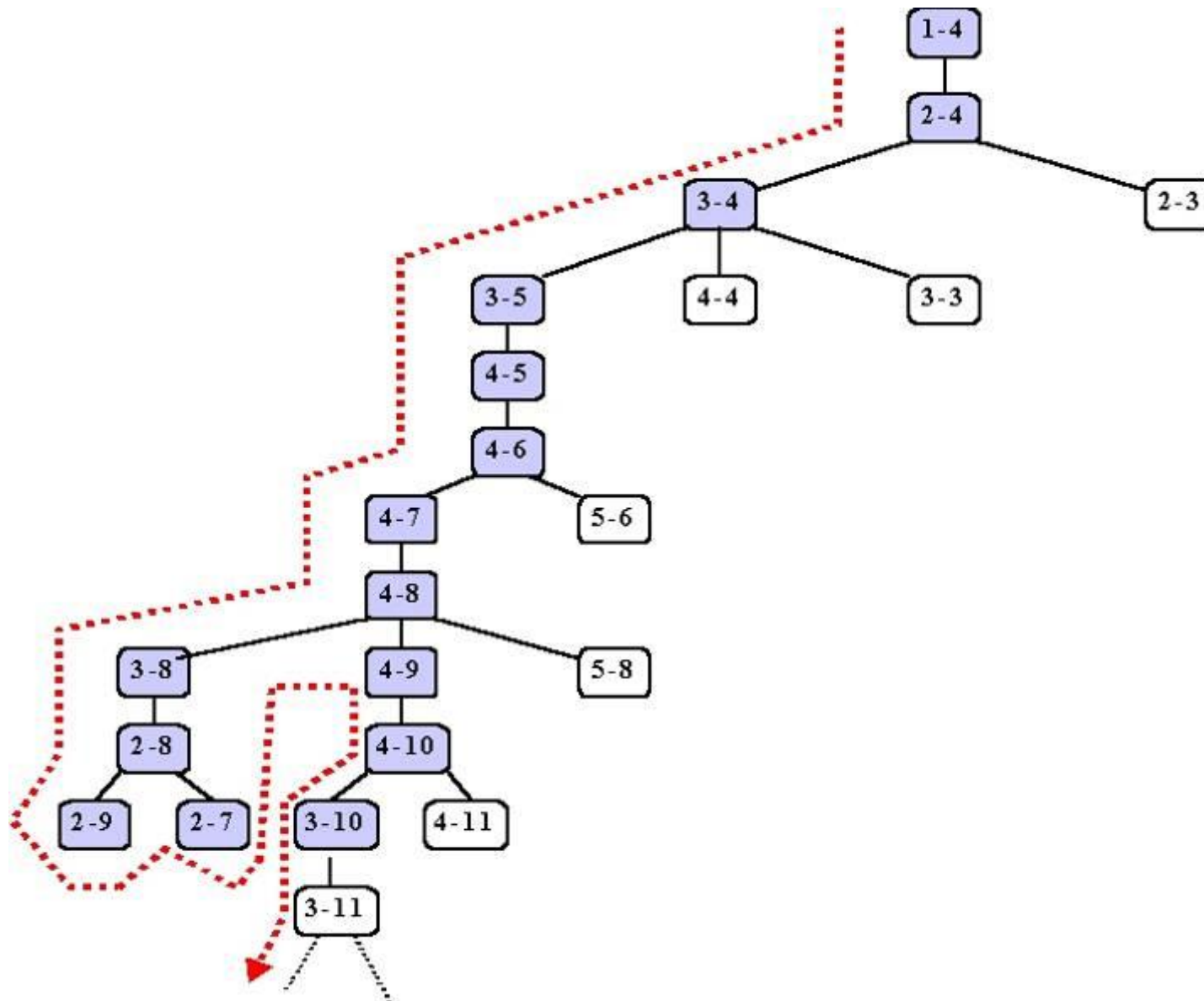
Εφαρμογή των Αλγορίθμων Τυφλής Αναζήτησης

Το πρόβλημα του Λαβύρινθου - Ορισμός

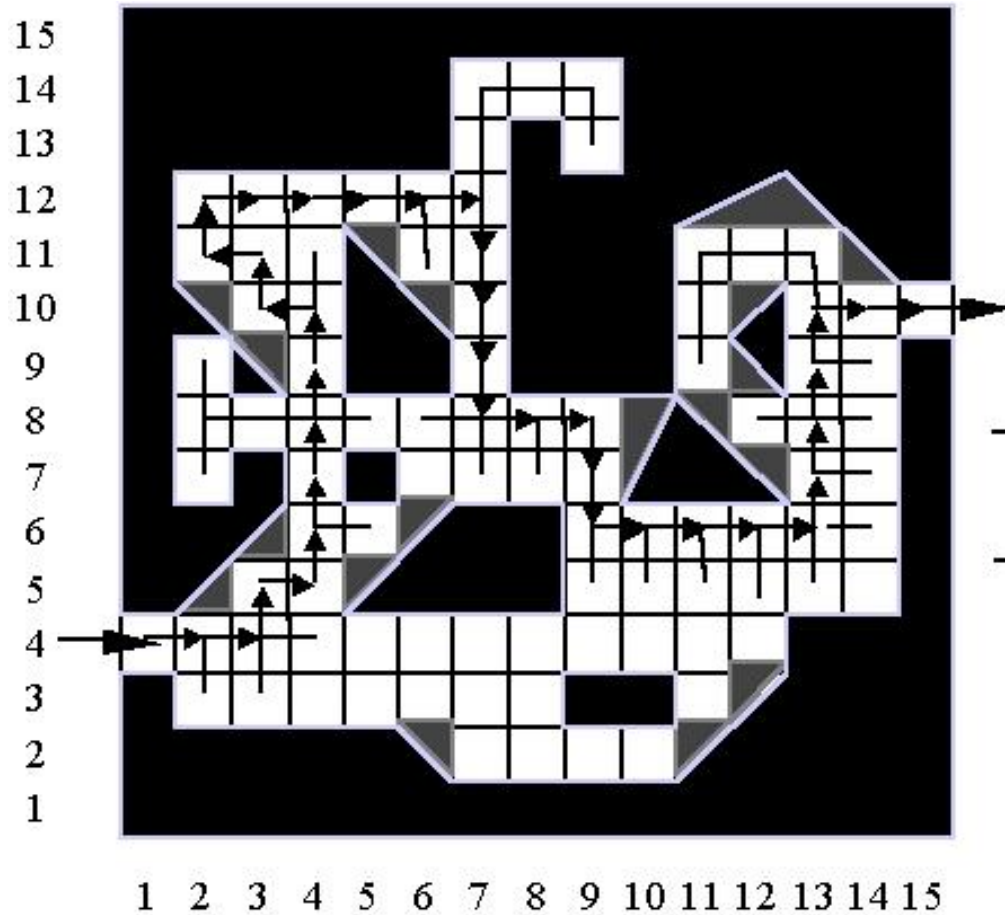


- ❖ Αρχική κατάσταση (1,4).
- ❖ Το σύνολο τελικών καταστάσεων περιέχει μόνο (15,10).
- ❖ Οι τελεστές μεταφοράς είναι οι εξής: πήγαινε μία θέση αριστερά, πήγαινε μία θέση επάνω, πήγαινε μία θέση δεξιά, πήγαινε μία θέση κάτω, εφόσον η θέση είναι ελεύθερη.
- ❖ Ο χώρος καταστάσεων είναι όλες οι ελεύθερες θέσεις, χωρίς εμπόδια, του πλέγματος.

Εφαρμογή του αλγορίθμου DFS

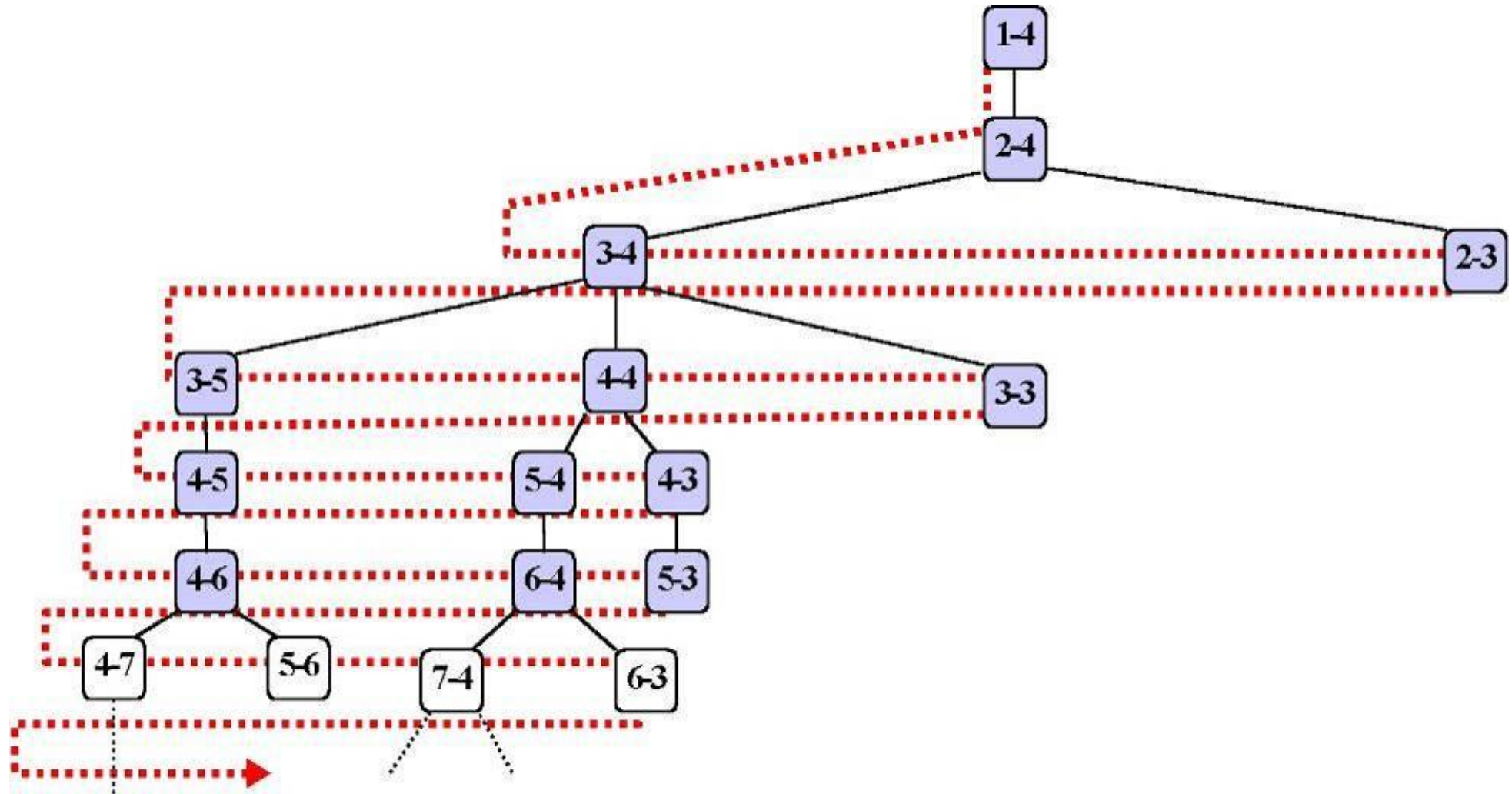


Λύση στο πρόβλημα του λαβύρινθου με χρήση DFS

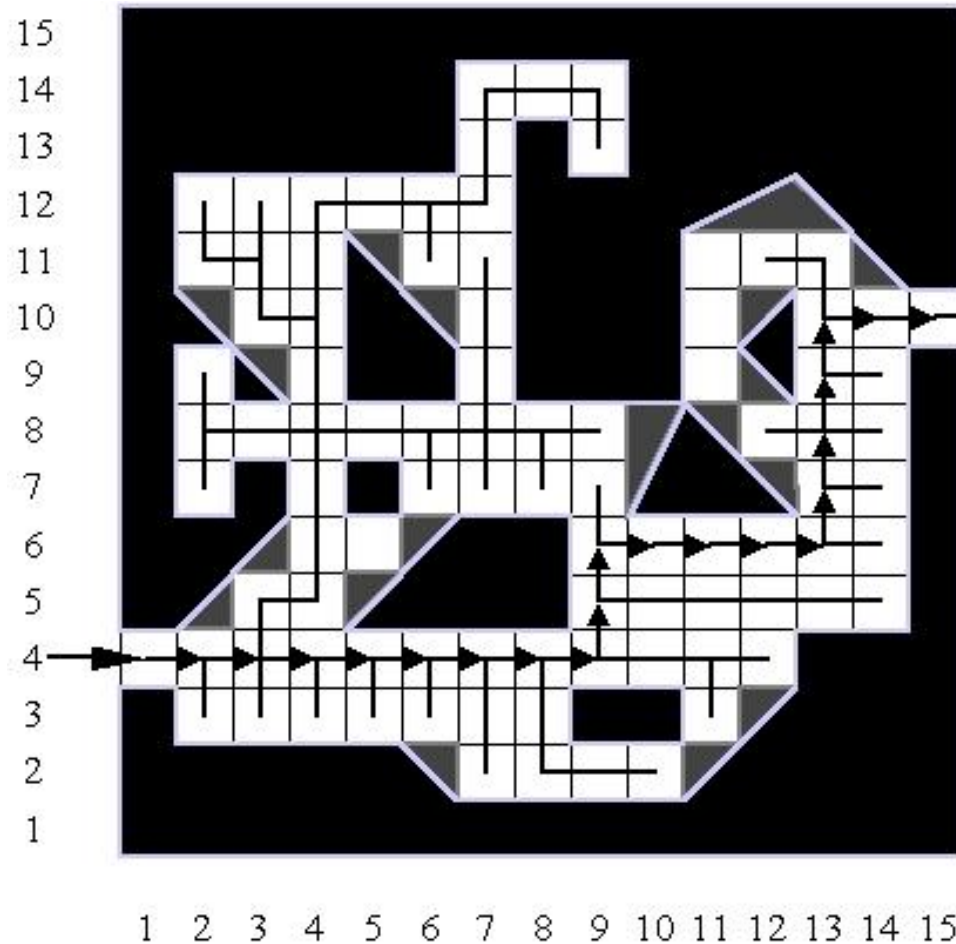


- Τελεστής που ανήκει στη λύση
- Τελεστής που δεν ανήκει στη λύση αλλά εξετάζεται κατά την εκτέλεση του αλγορίθμου

Εφαρμογή αλγορίθμου BFS



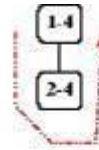
Λύση στο πρόβλημα του λαβύρινθου με χρήση BFS



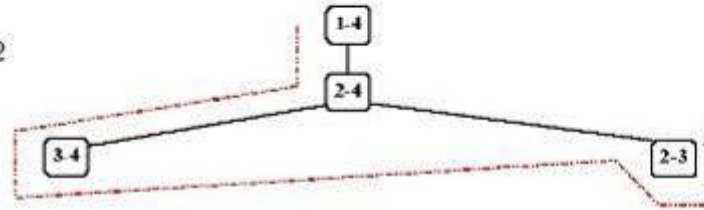
- Τελεστής που ανήκει στη λύση
- Τελεστής που δεν ανήκει στη λύση αλλά εξετάζεται κατά την εκτέλεση του αλγορίθμου

Εφαρμογή του ID στο πρόβλημα του λαβυρίνθου

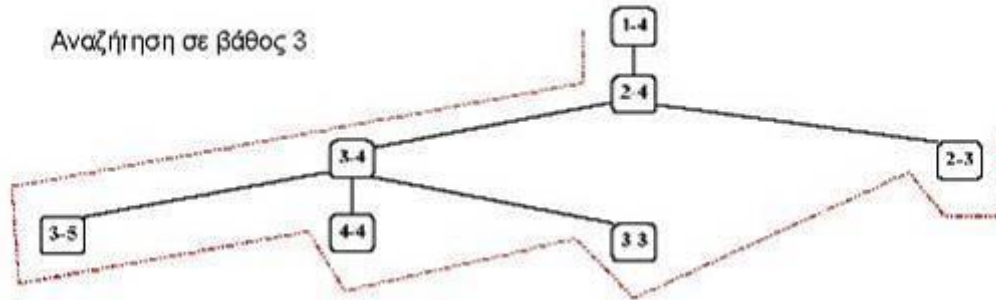
Αναζήτηση σε βάθος 1



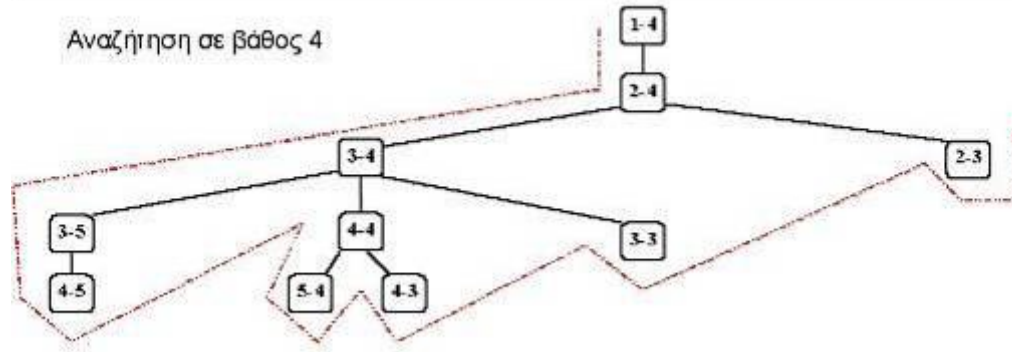
Αναζήτηση σε βάθος 2



Αναζήτηση σε βάθος 3



Αναζήτηση σε βάθος 4



Αλγόριθμοι Ευριστικής Αναζήτησης

Εισαγωγικά (1/2)

- ❖ Ο χώρος αναζήτησης συνήθως αυξάνεται εκθετικά. Απαιτείται πληροφορία για αξιολόγηση των καταστάσεων (ευριστικός μηχανισμός).
- ❖ Οι αλγόριθμοι που εκμεταλλεύονται τέτοια πληροφορία ονομάζονται **Αλγόριθμοι Ευριστικής Αναζήτησης**.
- ❖ Παράδειγμα ευριστικής αναζήτησης είναι η συναρμολόγηση ενός puzzle.
- ❖ Αν δεν υπήρχαν ευριστικοί μηχανισμοί, τότε τα προβλήματα αυτά θα λύνονταν πολύ δύσκολα, γιατί οι συνδυασμοί που πρέπει να γίνουν είναι πάρα πολλοί.
- ❖ Ο ευριστικός μηχανισμός εξαρτάται από τη γνώση που έχουμε για το πρόβλημα.

Ευριστικός μηχανισμός

Ευριστικός μηχανισμός (heuristic) είναι μία στρατηγική, βασισμένη στη γνώση για το συγκεκριμένο πρόβλημα, η οποία χρησιμοποιείται σα βοήθημα στη γρήγορη επίλυσή του.

- ❖ Ο ευριστικός μηχανισμός υλοποιείται με ευριστική συνάρτηση (heuristic function).
- ❖ Ευριστική τιμή (heuristic value) είναι η τιμή της ευριστικής συνάρτησης και εκφράζει το πόσο κοντά βρίσκεται μία κατάσταση σε μία τελική.
- ❖ Η ευριστική τιμή δεν είναι η πραγματική τιμή της απόστασης από μία τερματική κατάσταση, αλλά μία εκτίμηση (estimate) που πολλές φορές μπορεί να είναι και λανθασμένη.

1.2.7

Ευριστικές Συναρτήσεις σε Μικρά Προβλήματα (1/3)

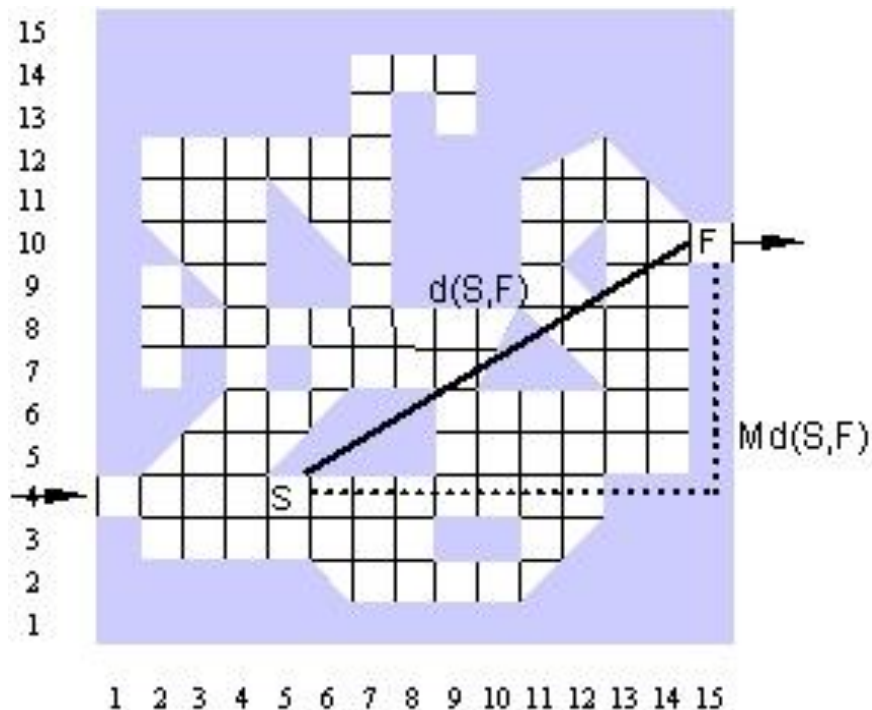
Ευριστικός μηχανισμός και συναρτήσεις σε λαβύρινθο

- ❖ Ευκλείδεια απόσταση (Euclidian distance):

$$d(S,F) = \sqrt{(X_S - X_F)^2 + (Y_S - Y_F)^2}$$

- ❖ Απόσταση Manhattan (Manhattan distance):

$$Md(S,F) = |X_S - X_F| + |Y_S - Y_F|$$



EMBED Equation.3

$$Md(S,F) = |5-15| + |4-10| = 10+6=16.$$

Ευριστικές Συναρτήσεις σε Μικρά Προβλήματα (2/3)

Ευριστικός μηχανισμός και συναρτήσεις στο N-Puzzle

- ❖ Πόσα πλακίδια βρίσκονται εκτός θέσης.
- ❖ Το άθροισμα των αποστάσεων Manhattan κάθε πλακιδίου από την τελική του θέση.

Αρχική κατάσταση

7	8	15	5
14	10	2	12
9	1	6	11
13	4	3	

Εκτός Θέσης = 12
Άθροισμα Manhattan
αποστάσεων = 28

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Τελική κατάσταση

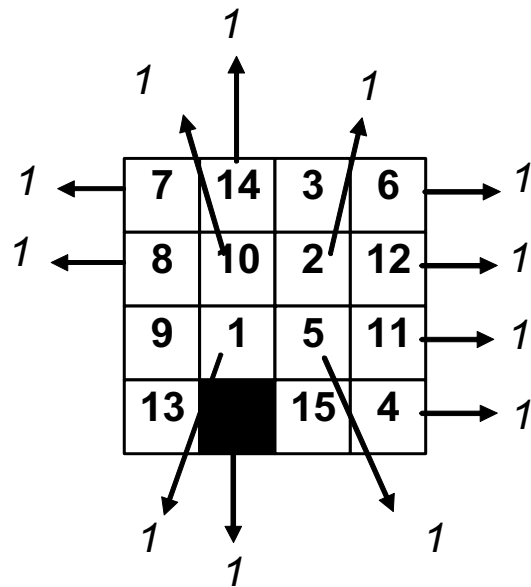
7	14	3	6
8	10	2	12
9	1	5	11
13		15	4

Ευριστική Τιμή
(εκτίμηση απόστασης)

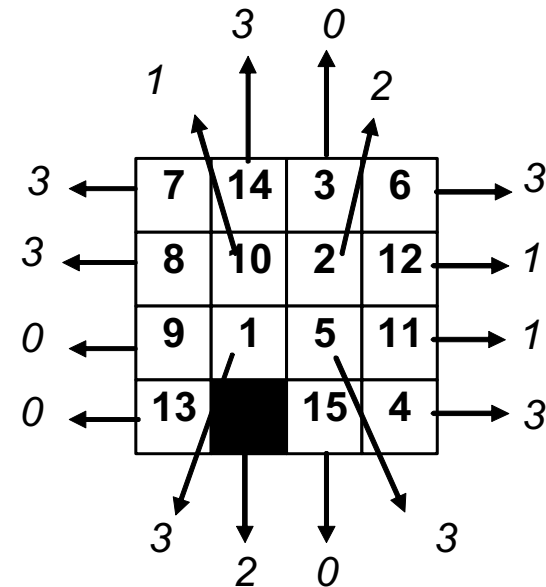
Τυχαία κατάσταση

Ευριστικές Συναρτήσεις σε Μικρά Προβλήματα (3/3)

Αναλυτικός υπολογισμός ευριστικής τιμής για μία τυχαία κατάσταση του 15-puzzle.



Εκτός θέσης = 12



Άθροισμα αποστάσεων Manhattan = 28

Ευριστικός μηχανισμός και συναρτήσεις στο TSP

- ❖ Η κοντινότερη πόλη έχει περισσότερες πιθανότητες να οδηγήσει σε μία συνολικά καλή λύση.

Αναζήτηση με Αναρρίχηση Λόφων

Η αναρρίχηση λόφων (Hill-Climbing Search - HC) είναι ένας αλγόριθμος αναζήτησης που μοιάζει πολύ με τον DFS.

Ο αλγόριθμος HC

1. Η αρχική κατάσταση είναι η τρέχουσα κατάσταση.
2. Αν η κατάσταση είναι μία τελική τότε ανάφερε τη λύση και σταμάτησε.
3. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
4. Βρες την καλύτερη κατάσταση σύμφωνα με την ευριστική συνάρτηση.
5. Η καλύτερη κατάσταση γίνεται η τρέχουσα κατάσταση.
6. Πήγαινε στο βήμα 2.

Ο αλγόριθμος HC (Ψευδοκώδικας)

```
algorithm hc(InitialState, FinalState)
begin
  CurrentState ← InitialState;
  while CurrentState ≠ FinalState do
    Children ← Expand(CurrentState);
    if Children = ∅ then return failure;
    EvaluatedChildren ← Heuristic(Children);
    bestChild ← best(EvaluatedChildren);
    if hValue(CurrentState) ≥ hValue(bestChild)
      then return failure;
      else CurrentState ← bestChild;
    endif;
  endwhile;
  return success;
end.
```

Ο αλγόριθμος HC

Σχόλια (1/2)

❖ Ο HC χρησιμοποιείται σε προβλήματα όπου πρέπει να βρεθεί μία λύση πολύ γρήγορα, έστω και αν αυτή δεν είναι η καλύτερη.

❖ Πλεονεκτήματα:

Πολύ αποδοτικός και σε χρόνο και σε μνήμη,

❖ Μειονεκτήματα:

Είναι μη-πλήρης.

Βασικά προβλήματα του HC:

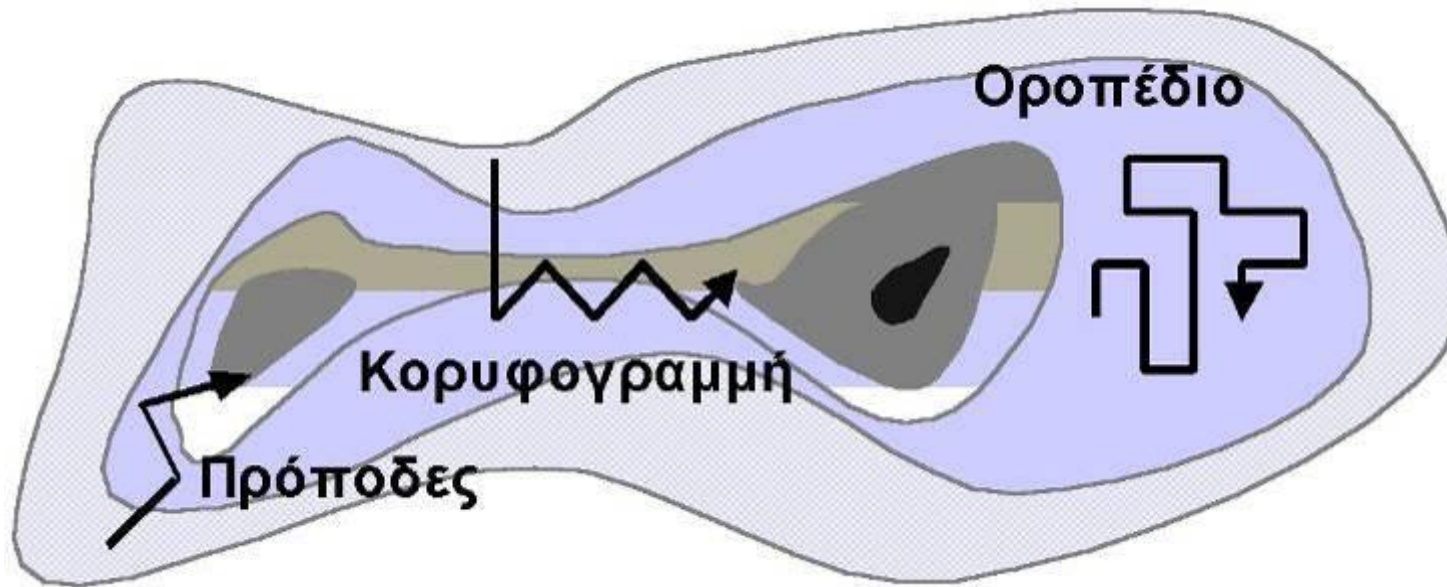
✓ Πρόποδες (foothill).

✓ Οροπέδιο (plateau).

✓ Κορυφογραμμή (ridges).

Ο αλγόριθμος HC

Σχόλια (2/2)



❖ Βελτιώσεις:

- Εξαναγκασμένη αναρρίχηση λόφου (*Enforced Hill-Climbing - EHC*)
- Προσομοιωμένη εξέλιξη (*Simulated Annealing - SA*)
- Αναζήτηση με απαγορευμένες καταστάσεις (*Tabu Search - TS*).

1.2.9

Ακτινωτή Αναζήτηση

Στον αλγόριθμο ακτινωτής αναζήτησης (*Beam Search - BS*) δεν κλαδεύονται όλες οι υπόλοιπες καταστάσεις όπως στον HC, αλλά ένας σταθερός αριθμός από τις καλύτερες από αυτές κρατείται στο μέτωπο αναζήτησης.

1.2.10

Αναζήτηση Πρώτα στο Καλύτερο

Ο αλγόριθμος αναζήτηση πρώτα στο καλύτερο (Best-First - *BestFS*) κρατά όλες τις καταστάσεις στο μέτωπο αναζήτησης.

Ο αλγόριθμος BestFS

1. Βάλτε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Πάρτε την πρώτη σε σειρά κατάσταση από το μέτωπο αναζήτησης.
4. Αν η κατάσταση είναι μέλος του κλειστού συνόλου τότε πήγαινε στο 2.
5. Αν η κατάσταση είναι μία τελική τότε ανάφερε τη λύση και σταμάτα.
6. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά.
7. Εφάρμοσε την ευριστική συνάρτηση σε κάθε παιδί.
8. Βάλτε τις καταστάσεις-παιδιά στο μέτωπο αναζήτησης.
9. Αναδιάταξε το μέτωπο αναζήτησης, έτσι ώστε η κατάσταση με την καλύτερη ευριστική τιμή να είναι πρώτη.

10. Βάλτε τη κατάσταση-γονέα στο κλειστό σύνολο.
11. Πήγαινε στο βήμα 2.

Ο αλγόριθμος BestFS (Ψευδοκώδικας)

```
algorithm bestfs (InitialState, FinalStates)
begin
  Closed $\emptyset$ ;
  EvaluatedInitialStateHeuristic (<InitialState>)
  Frontier<EvaluatedInitialState>;
  CurrentStatebest (Frontier);
  while CurrentState  $\notin$  FinalStates do
    Frontierdelete (CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet then
      begin
        Children Expand (CurrentState);
        EvaluatedChildrenHeuristic (Children);
        FrontierFrontier  $\wedge$  EvaluatedChildren;
        ClosedClosed {CurrentState};
      end;
    if Frontier=  $\emptyset$  then return fail;
    CurrentStatebest (Frontier);
  endwhile;
  return success;
end.
```

Ο αλγόριθμος BestFS

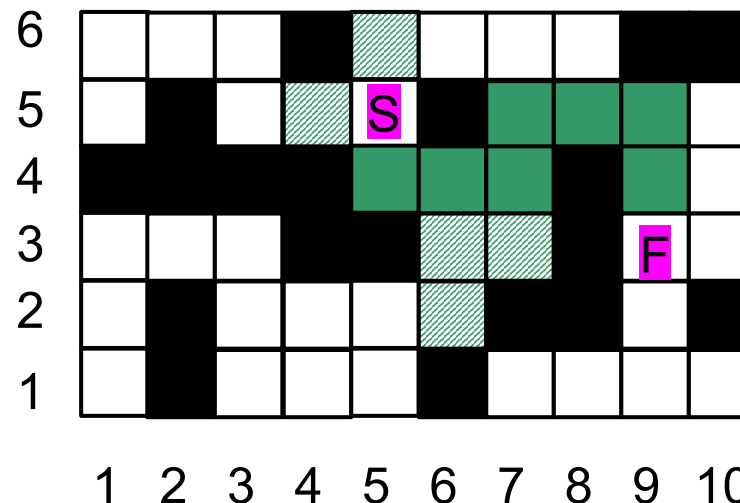
Σχόλια

❖ Πλεονεκτήματα:

- ❑ Προσπαθεί να δώσει μια γρήγορη λύση σε κάποιο πρόβλημα. Εξαρτάται πολύ από τον ευριστικό μηχανισμό.
- ❑ Είναι πλήρης.

❖ Μειονεκτήματα:

- ❑ Το μέτωπο αναζήτησης μεγαλώνει με υψηλό ρυθμό και μαζί του ο χώρος που χρειάζεται για την αποθήκευσή του.
- ❑ Δεν εγγυάται ότι η λύση που θα βρεθεί είναι η βέλτιστη.



Ο αλγόριθμος BestFS: το πρόβλημα του λαβύρινθου

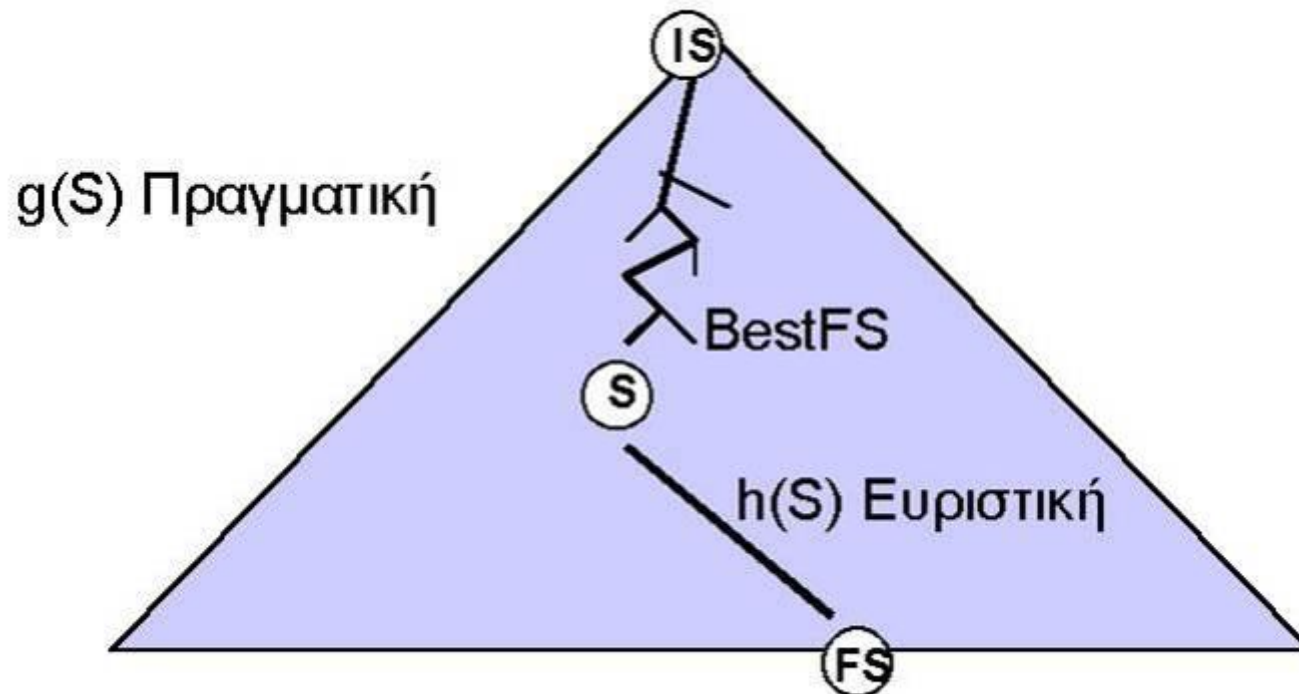
Μέτωπο Αναζήτησης	Κλειστό Σύνολο	Κατ.	Παιδιά
<5-5>	<>	5-5	5-45,5-67,4-57
<5-45,5-67,4-57>	<5-5>	5-4	5-56,6-44
<6-44,5-56,5-67,4-57>	<5-5,5-4>	6-4	5-45,6-33,7-43
<7-43,6-33,6-44,5-45,5-56,5-67,4-57>	<5-5,5-4,6-4>	7-4	7-54, 7-32, 6-44
<7-32, 6-33, 7-54,6-44,5-45,5-56,5-67,4-57>	<5-5,5-4,6-4,7-4>	7-3	6-33, 7-43
<6-33,7-43,7-54,...>	<5-5,5-4,6-4,7-4,7-3>	6-3	6-44,6-24,7-32
<7-32,7-43,6-24,6-44,7-54,...>	<...,6-3>	7-3	Βρόχος
<7-43,6-24, 6-44,7-54,...>	<...>	7-4	Βρόχος
<6-24, 6-44,7-54,...>	<...>	6-2	5-25,6-33
<6-32,6-24, 6-44,7-54,...>	<...,6-2>	6-3	Βρόχος
<6-24, 6-44,7-54,...>	<...>	6-4	Βρόχος
<7-54,5-25,...>	<...>	7-5	8-53,7-65,7-43
<8-53,7-43,7-65,...>	<...,7-5>	8-5	8-64,7-54,9-52
<9-52, 8-53,7-43,...>	<...,8-5>	9-5	8-53,9-41,10-53
<9-41,10-53,8-53,7-43,...>	<...,9-5>	9-4	9-30,9-52,10-42
<9-30,...>	<...,9-4>	9-3	ΤΕΛΙΚΗ ΚΑΤΑΣΤΑΣΗ

Ο Αλγόριθμος Άλφα-Άστρο (A*)

Ο αλγόριθμος A (Άλφα Άστρο) είναι κατά βάσει BestFS, αλλά με ευριστική συνάρτηση:

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

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 ευριστικής συνάρτησης, όπως ακριβώς στον BestFS.



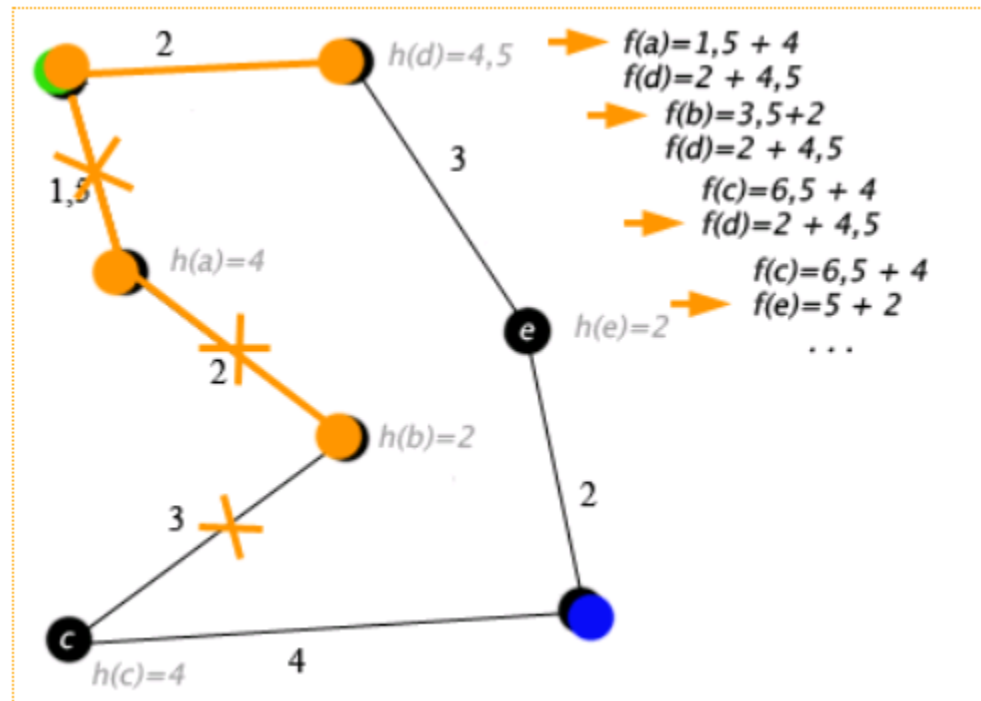
Ο Αλγόριθμος Άλφα-Άστρο (A^*)

Σχόλια

- ❖ 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 ο A^* βρίσκει πάντα τη βέλτιστη λύση.
- ❖ Στην περίπτωση αυτή ο ευριστικός μηχανισμός ονομάζεται αποδεκτός (admissible)
- ❖ Βελτιώσεις:
 - A^* με επαναληπτική εκβάθυνση (Iterative Deepening A^* - IDA)

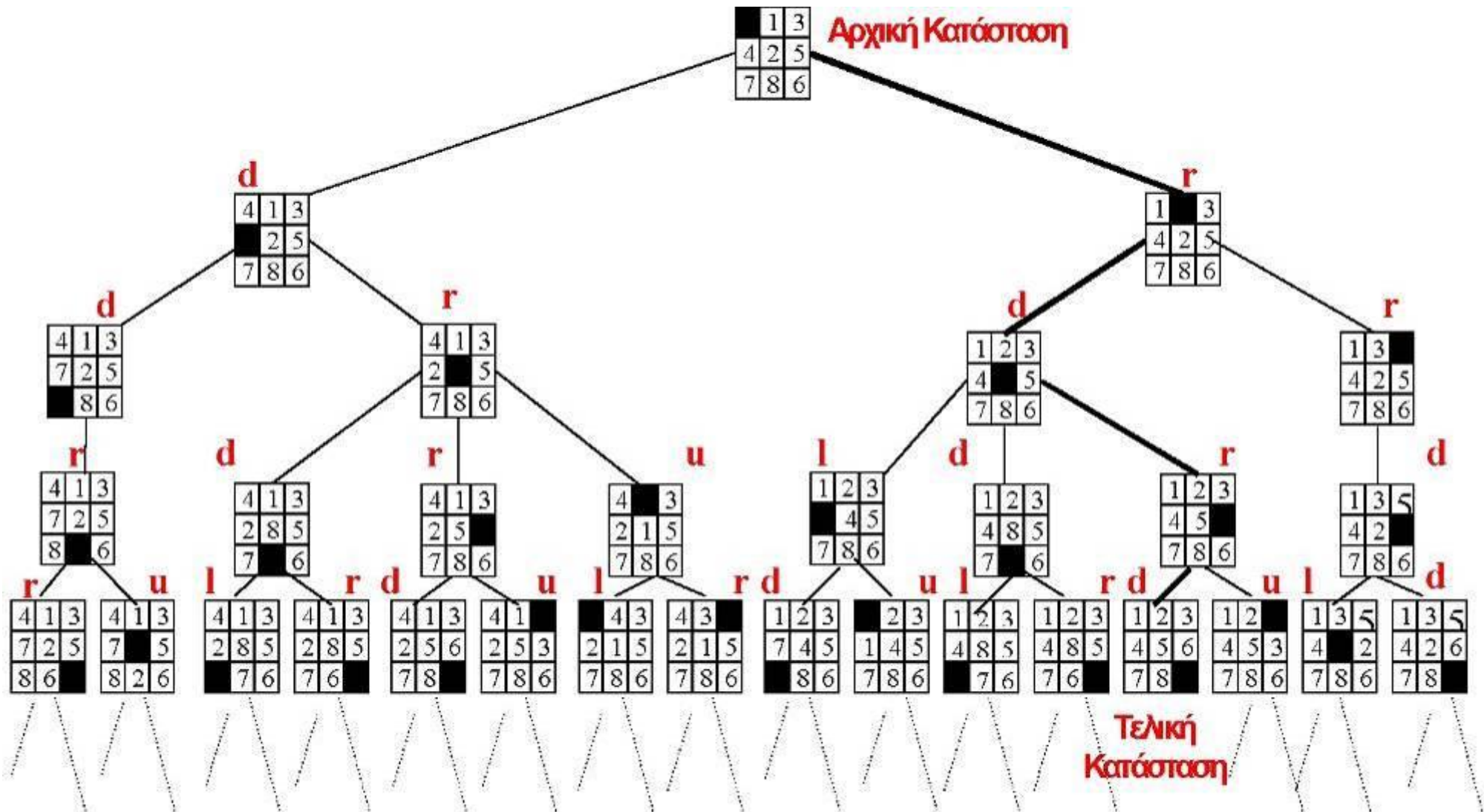
1.2.11

Παράδειγμα A*

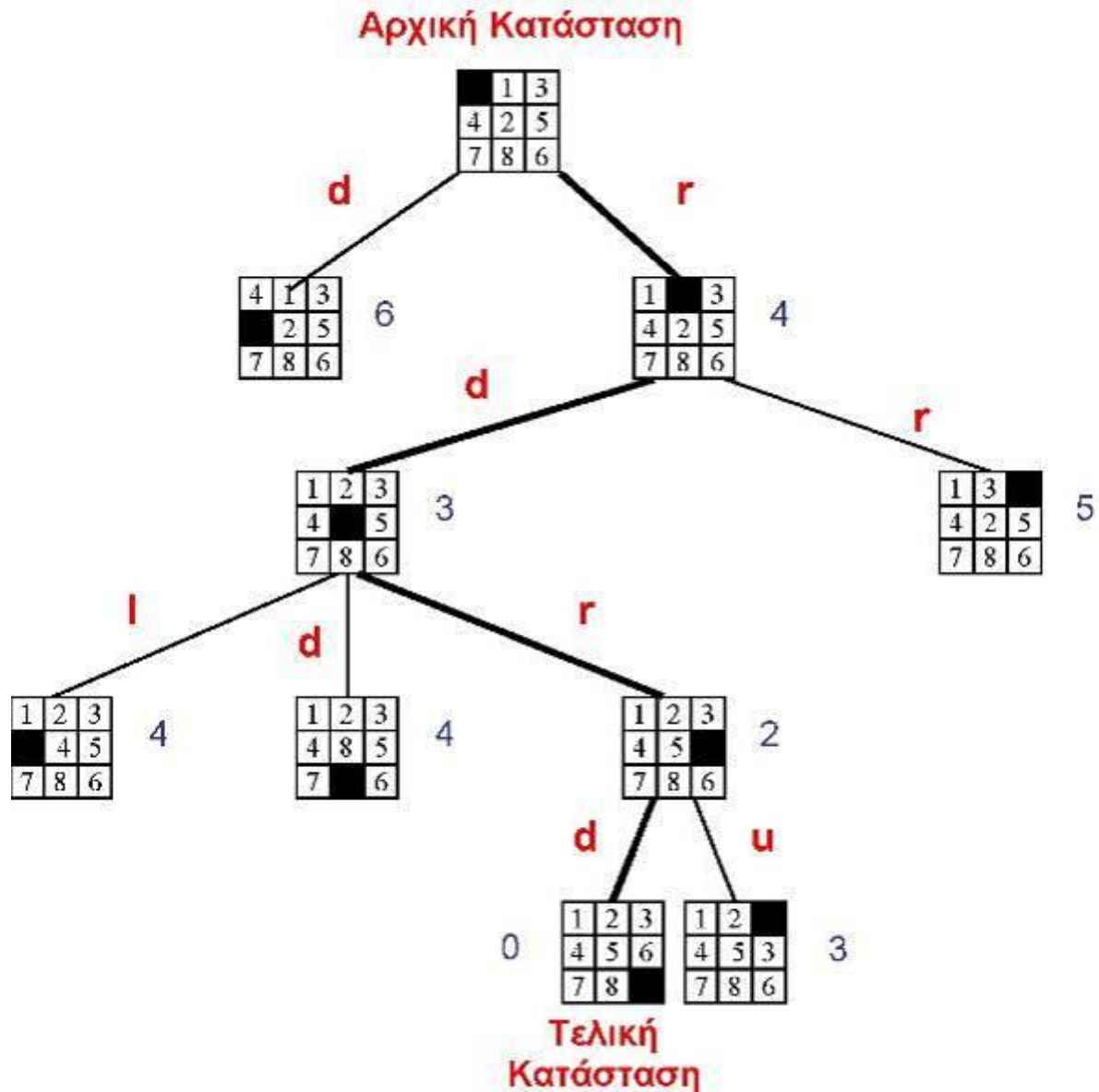


Εφαρμογή των Αλγορίθμων Ευριστικής Αναζήτησης

Χώρος Αναζήτησης στο 8-puzzle



Εφαρμογή αλγορίθμου BestFS στο 8-puzzle



Αλγόριθμοι Αναζήτησης σε Παίγνια Δύο Αντιπάλων

Εισαγωγικά (1/3)

- ❖ Τα προβλήματα όπου η εξέλιξη των καταστάσεων εξαρτάται από δύο διαφορετικά σύνολα τελεστών μετάβασης που εφαρμόζονται εναλλάξ αναφέρονται και ως ανταγωνιστικά παίγνια ή παίγνια δύο αντιπάλων (*adversary ή two-person games*)
- ❖ Ο όρος "παίγνιο" αφορά την περιγραφή του τρόπου με τον οποίο παίζεται το παιχνίδι και περιλαμβάνει:
 - ❑ τα αντικείμενα που υπάρχουν (για παράδειγμα, τα πούλια, το ταμπλώ, κτλ.) καθώς και
 - ❑ το σύνολο των κανόνων που το διέπουν.
- ❖ Αντίθετα, με τον όρο "παιχνίδι" χαρακτηρίζεται μία συγκεκριμένη παρτίδα του παιχνιδιού.

Αλγόριθμοι Αναζήτησης σε Παίγνια Δύο Αντιπάλων

Εισαγωγικά (2/3)

- ❖ Το πρόβλημα ορίζεται ως εξής:
 - ❑ Μια κατάσταση παριστάνει τη διάταξη των πιονιών σε κάποια χρονική στιγμή.
 - ❑ Ο χώρος καταστάσεων αποτελείται από όλες αυτές τις πιθανές επιτρεπτές καταστάσεις.
 - ❑ Οι τελεστές μετάβασης είναι οι επιτρεπτές κινήσεις (κανόνες του παιχνιδιού).
 - ❑ Οι τελικές καταστάσεις έχουν γνωστά χαρακτηριστικά (π.χ. ματ στο σκάκι).

- ❖ Έστω ότι κάποιος είναι η σειρά του να κάνει μία κίνηση.
 - ❑ Αν θέλει να κερδίσει, θα επιδιώξει να κάνει την καλύτερη κίνηση για αυτόν, η οποία αντίστοιχα θα είναι και η χειρότερη για τον αντίπαλο.
 - ❑ Ο τρόπος που σκέφτεται είναι ο εξής: "Αν κάνω αυτή την κίνηση, τότε ο αντίπαλος θα κάνει εκείνη, αν όμως κάνω την άλλη κίνηση, τότε θα κάνει αυτήν, ..." κ.ο.κ.
 - ❑ Ο συλλογισμός αυτός αντιστοιχεί στη δημιουργία ενός δένδρου (**δένδρο του παιχνιδιού game**).

Αλγόριθμοι Αναζήτησης σε Παίγνια Δύο Αντιπάλων

Εισαγωγικά (3/3)

- ❖ Το χαρακτηριστικό του είναι ότι οι κινήσεις δύο διαδοχικών επιπέδων ανήκουν σε διαφορετικό παίκτη, γιατί οι παίκτες παίζουν εναλλάξ.

- ❖ Έστω ότι ένας από τους δύο αντιπάλους είναι ο υπολογιστής.
 - ❑ Το πρόγραμμα αναζήτησης πρέπει να εξετάσει όλες τις πιθανές κινήσεις που παράγονται από μία (αρχική) κατάσταση.
 - ❑ Ένα τέτοιο πρόγραμμα μπορεί να προβλέψει την έκβαση του παιχνιδιού μετά από 10 κινήσεις (ο άνθρωπος σταματά συνήθως μετά την πρόβλεψη 2 ή 3 κινήσεων).
- ❖ Παραδόξως, οι διαφορές αυτές δεν κάνουν κάποιο πρόγραμμα κυρίαρχο οποιουδήποτε παιχνιδιού έναντι του ανθρώπου.
- ❖ Η ανωτερότητα των ανθρώπων έγκειται στους εξής παράγοντες:
 - ❑ Οι άνθρωποι επιλέγουν με κάποιον ευριστικό τρόπο τις εναλλακτικές κινήσεις.
 - ❑ Οι άνθρωποι διαθέτουν κάποια διαίσθηση για την κατάληξη του παιχνιδιού.
 - ❑ Με την εμπειρία που αποκτούν, μπορούν να σκέφτονται πολλές φορές εντελώς μηχανικά, ιδίως στα πρώτα και τελευταία στάδια του παιχνιδιού.

Ο Αλγόριθμος Minmax (1/3)

- ❖ Δεδομένης μίας κατάστασης του παιχνιδιού, ο αλγόριθμος αναζήτησης μεγίστου-ελαχίστου (*Minimax*) καλείται να αποφασίσει ποια θα είναι η επόμενη κίνησή του έναντι του αντιπάλου.
- ❖ Η εξαντλητική αναζήτηση των δένδρων αναζήτησης είναι ανέφικτη.
- ❖ Το ζητούμενο είναι:
 - Να χτιστεί το δένδρο μέχρι κάποιο βάθος
 - Να βρεθεί η καλύτερη κίνηση από την παρούσα κατάσταση.
- ❖ Το μέτρο της υπεροχής του ενός ή του άλλου αντιπάλου δίνεται από μία συνάρτηση αξιολόγησης (evaluation function) και η οποία εφαρμόζεται στα φύλλα του δένδρου του παιχνιδιού.
- ❖ Ο ένας παίκτης (πρόγραμμα) ονομάζεται max και ο άλλος (άλλο πρόγραμμα ή άνθρωπος) ονομάζεται min.

Αλγόριθμος Minimax

Ψευδογλώσσα

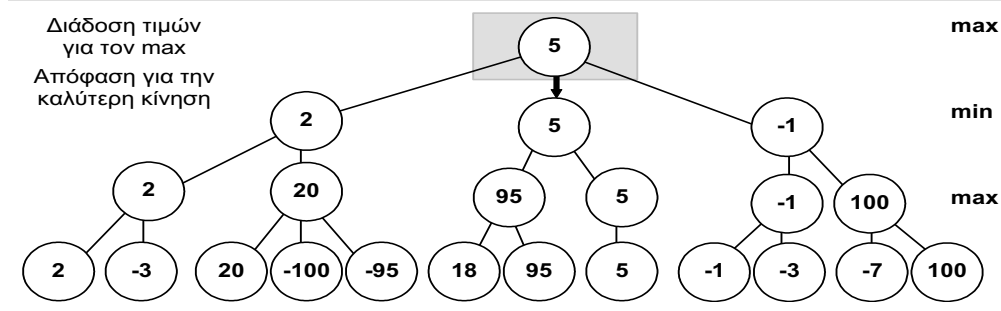
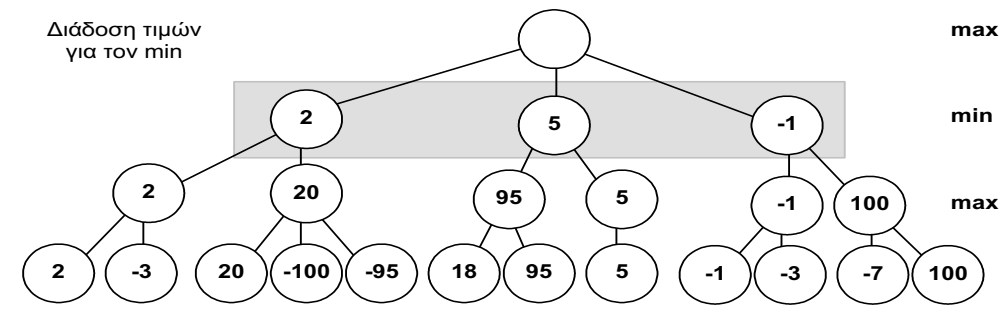
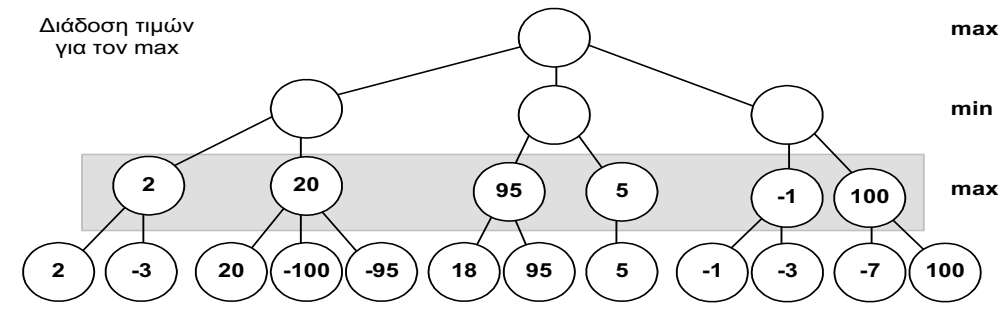
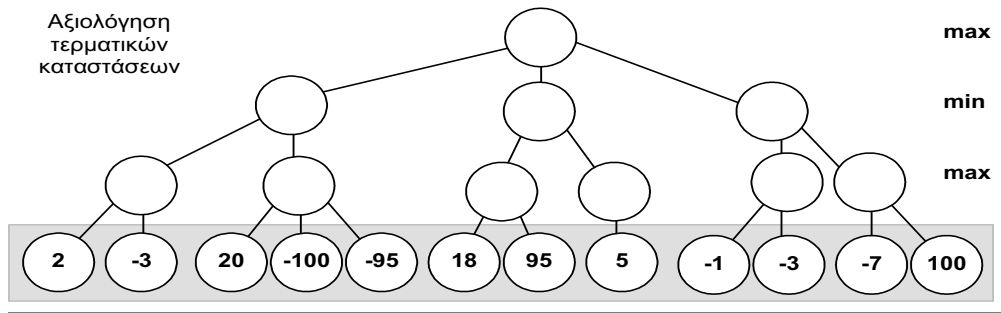
1. Εφάρμοσε τη συνάρτηση αξιολόγησης σε όλους τους κόμβους-φύλλα του δένδρου.
2. Έως ότου η ρίζα του δένδρου αποκτήσει τιμή, επανέλαβε:
3. Αρχίζοντας από τα φύλλα του δένδρου και προχωρώντας προς τη ρίζα, μετέφερε τις τιμές προς τους ενδιαμέσους κόμβους του δένδρου ως εξής:
 - i. Η τιμή κάθε κόμβου Max είναι η μέγιστη (maximum) των τιμών των κόμβων-παιδιών του.
 - ii. Η τιμή κάθε κόμβου Min είναι η ελάχιστη (minimum) των τιμών των κόμβων-παιδιών του.
4. Καλύτερη κίνηση είναι η κίνηση που οδηγεί στον κόμβο που έδωσε την πιο συμφέρουσα στη ρίζα τιμή (μέγιστη για το Max, ελάχιστη για το Min).

❖ Ο αλγόριθμος εγγυάται την πιο συμφέρουσα εξέλιξη μετά από κάποιες κινήσεις, έστω και αν ο αντίπαλος διαλέγει τις καλύτερες για αυτόν κινήσεις.

Ο Αλγόριθμος Minimax (2/3)

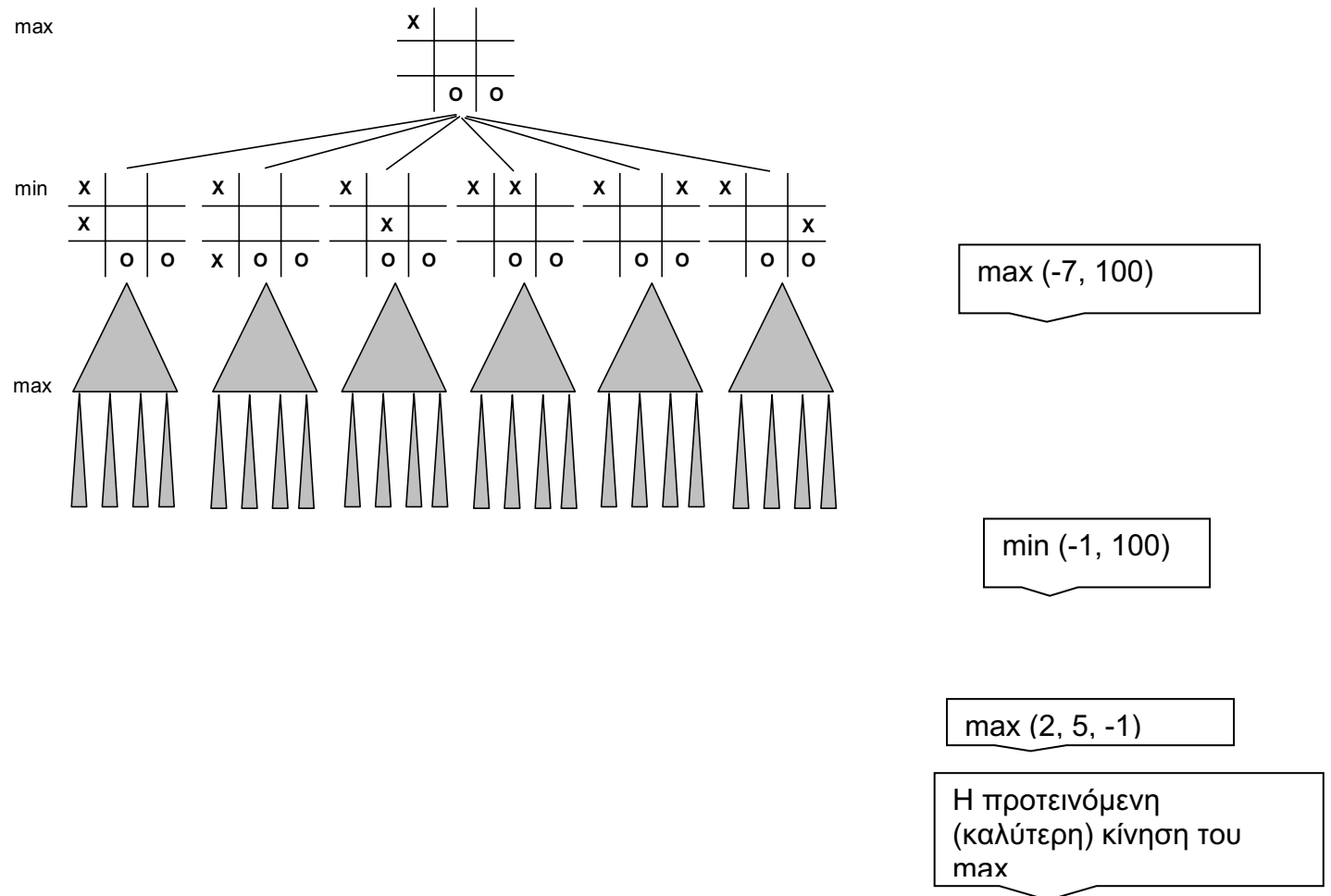
- ❖ Κατά σύμβαση, ο παίκτης που βρίσκεται στη ρίζα θεωρείται πως είναι ο Max.
- ❖ Οι καταστάσεις-φύλλα ονομάζονται τερματικές καταστάσεις, όμως δεν είναι απαραίτητα τελικές καταστάσεις.
- ❖ Οι τιμές των τερματικών καταστάσεων υπολογίζονται από τη συνάρτηση αξιολόγησης ενώ οι άλλες προκύπτουν από τη διάδοση αυτών.

Ο Αλγόριθμος Minimax (3/3)



Ο Αλγόριθμος Minimax στην Τρίλιζα (1/3)

- ❖ Η τρίλιζα έχει μικρό χώρο αναζήτησης ($9!$ καταστάσεις).

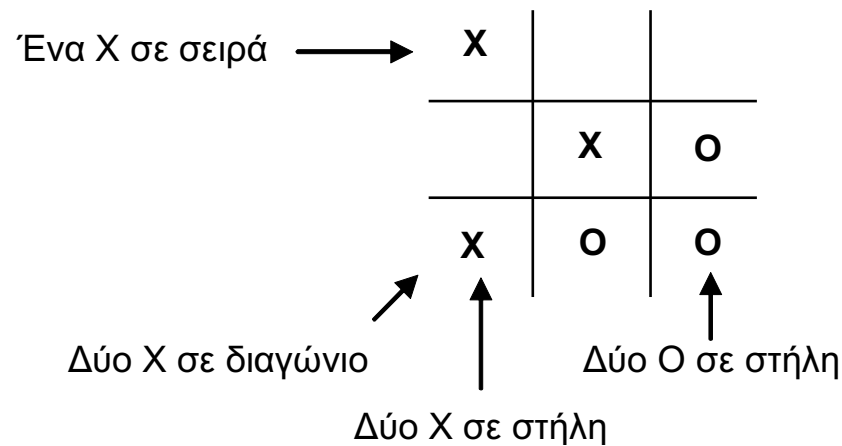


Ο Αλγόριθμος Minimax στην Τρίλιζα (2/3)

❖ Μία συνάρτηση αξιολόγησης στην τρίλιζα θα μπορούσε να είναι η:

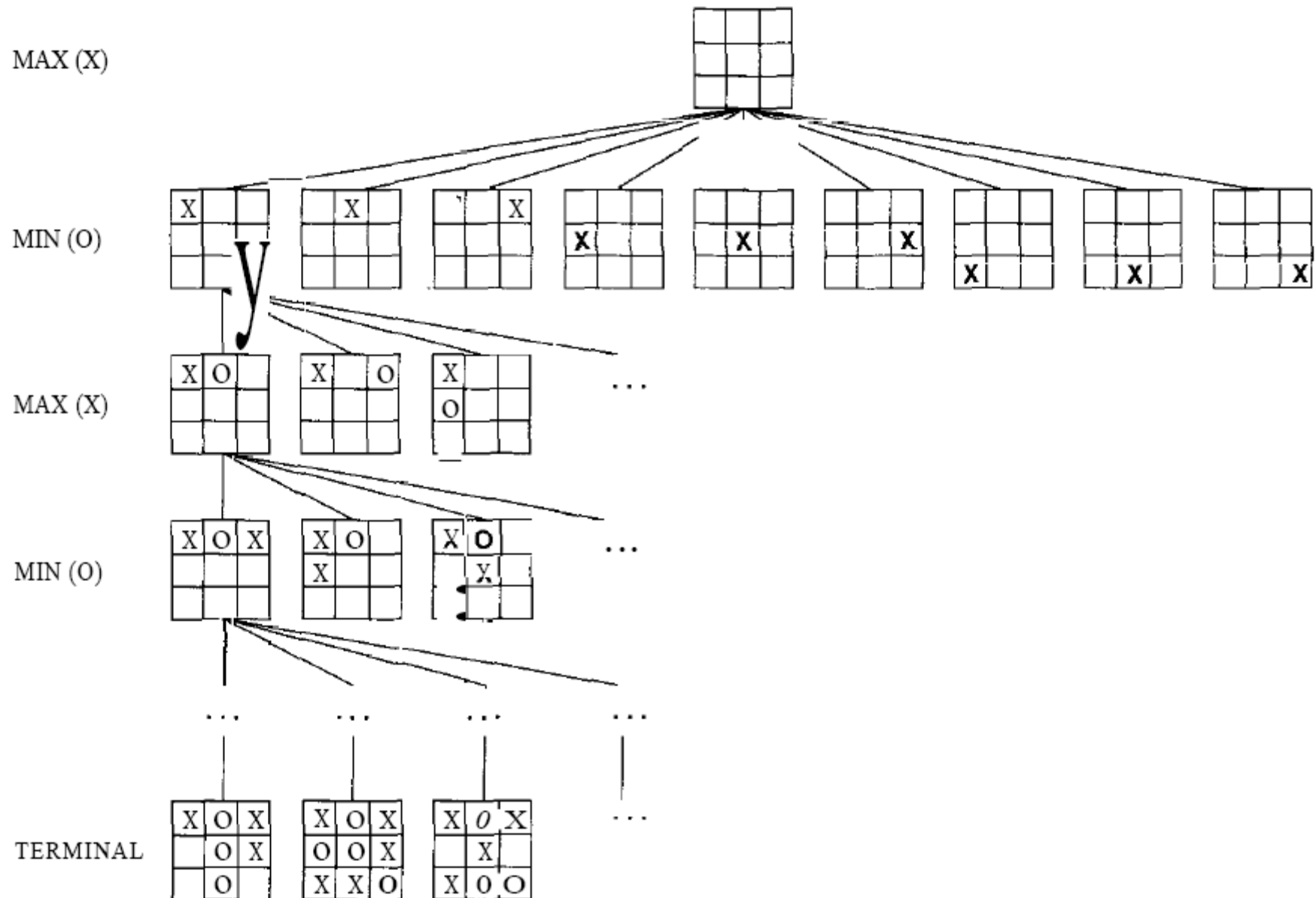
$$3 \cdot X_2 + X_1 - (3 \cdot O_2 + O_1)$$

- ❑ X_2 ο αριθμός γραμμών, στηλών ή διαγωνίων με δύο X και χωρίς κανένα O.
- ❑ X_1 ο αριθμός γραμμών, στηλών ή διαγωνίων με ένα X και χωρίς κανένα O.
- ❑ O_2 ο αριθμός γραμμών, στηλών ή διαγωνίων με δύο O και χωρίς κανένα X.
- ❑ O_1 ο αριθμός γραμμών, στηλών ή διαγωνίων με ένα O και χωρίς κανένα X.



$$3 \cdot X_2 + X_1 - (3 \cdot O_2 + O_1) = 3 \cdot 2 + 1 - (3 \cdot 1 + 0) = 4$$

Ο Αλγόριθμος Minimax στην Τρίλιζα (3/3)



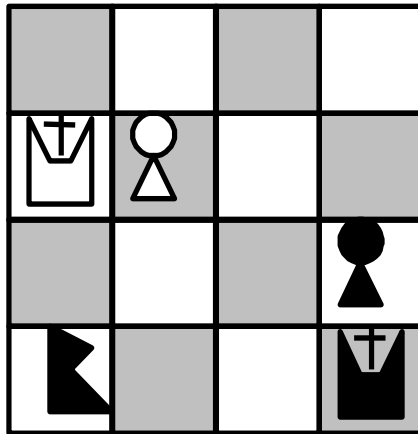
Εφαρμογή αλγορίθμου Minimax στο σκάκι

❖ Το κύριο μέλημα των προγραμμάτων σκάκι είναι να αναζητήσουν το δένδρο του παιχνιδιού σε όσο το δυνατόν μεγαλύτερο βάθος.

Μία συνάρτηση αξιολόγησης στο σκάκι

- ❑ (α) Υπεροχή κομματιών: π.χ. Βασιλιάς=10, Άλογο=5, Πιόνι=1 κλπ. Η αξία όλων των κομματιών κάθε χρώματος προστίθεται.
- ❑ (β) Υπεροχή θέσης: Κάθε κομμάτι που βρίσκεται στα 4 κεντρικά τετράγωνα παίρνει επιπλέον 2 πόντους.
- ❑ (γ) Απειλές: Για κάθε απειλή που προβάλλει ένας παίκτης παίρνει 3 επιπλέον πόντους, εκτός αν απειλεί το βασιλιά του άλλου παίκτη, οπότε παίρνει 20 πόντους.

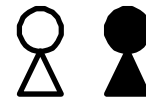
Εφαρμογή αλγορίθμου Minimax στο σκάκι (2/3)



Βασιλιάς



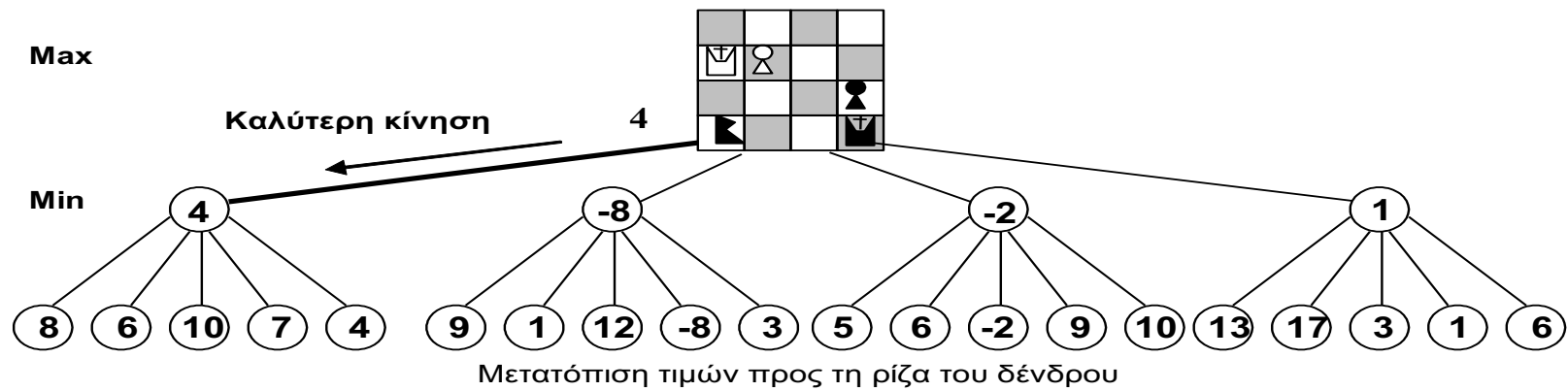
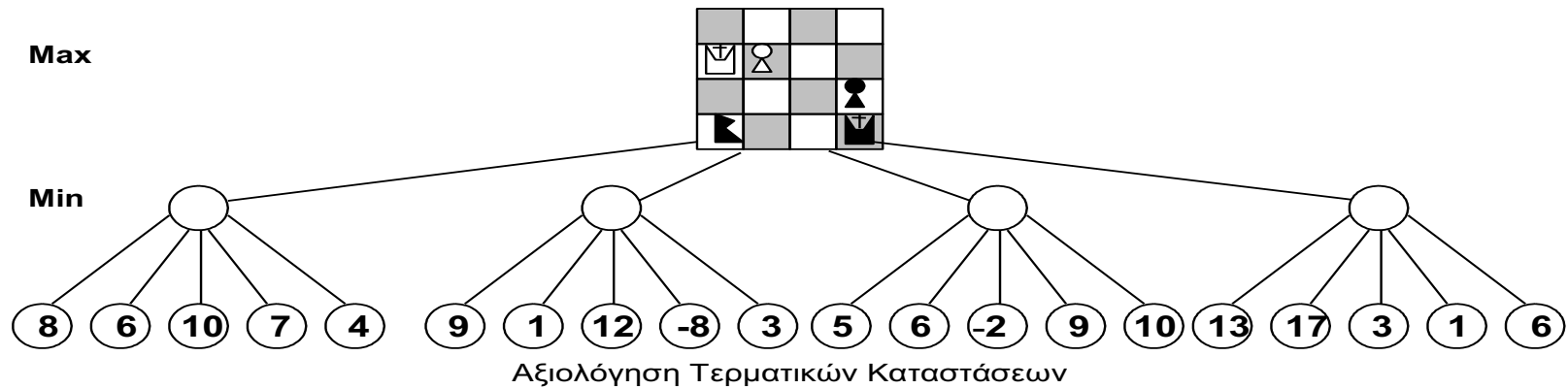
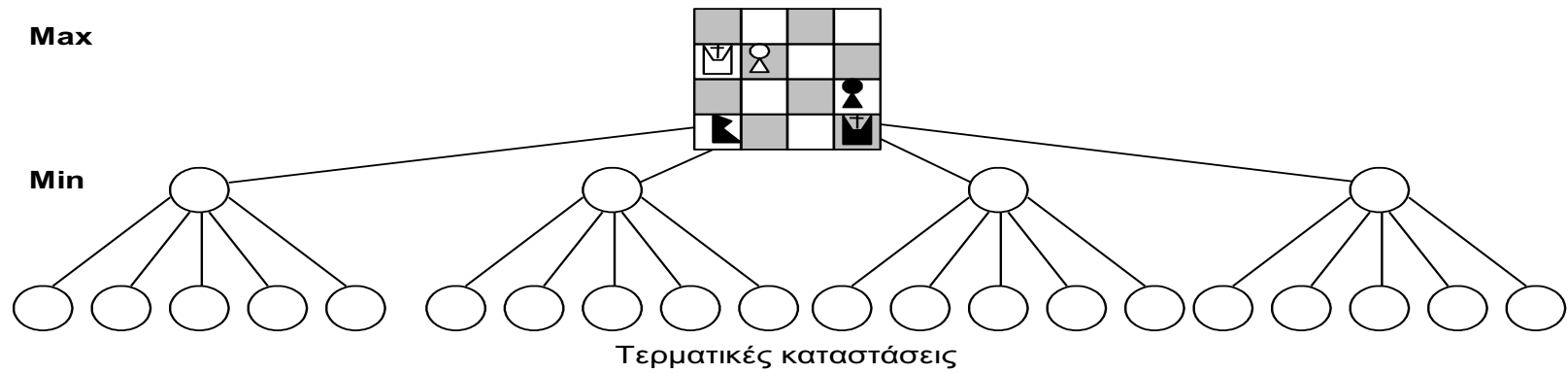
Άλογο



Πιόνι

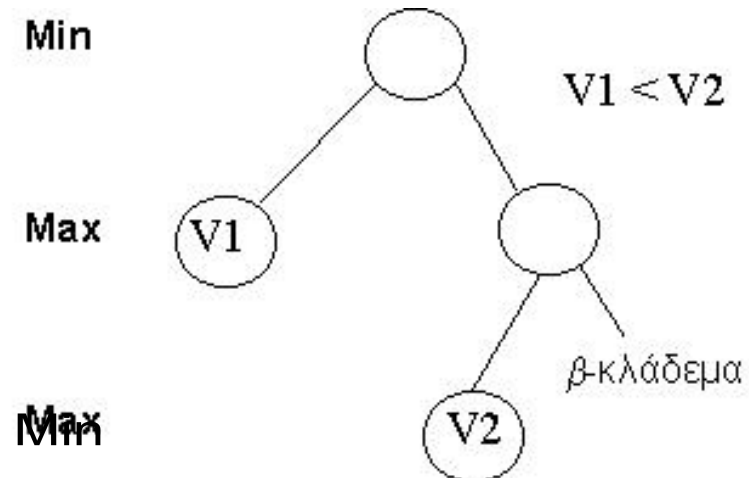
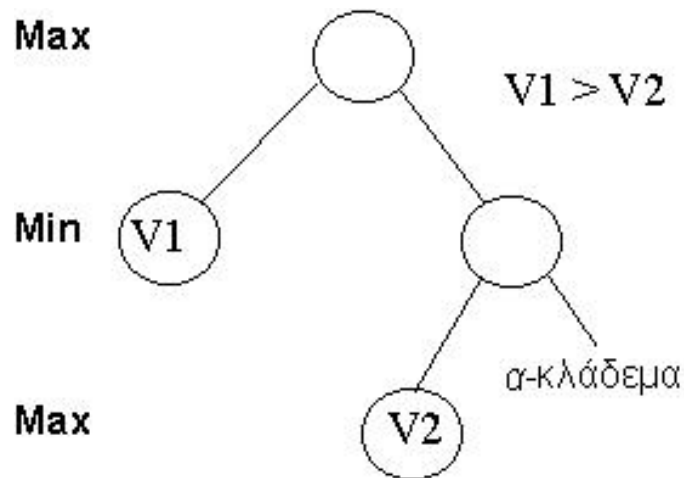
- ❖
- ❖ Λευκά : $10 + 1$ (λόγω α) + 2 (λόγω β) + 0 (λόγω γ) = 13
- ❖ Μαύρα : $10 + 5 + 1$ (λόγω α) + 0 (λόγω β) + 3 (λόγω γ) = 19
- ❖ Άρα τα μαύρα έχουν έχουν υπεροχή $19 - 13 = 6$

Ο Αλγόριθμος Minimax στο Σκάκι (3/3)



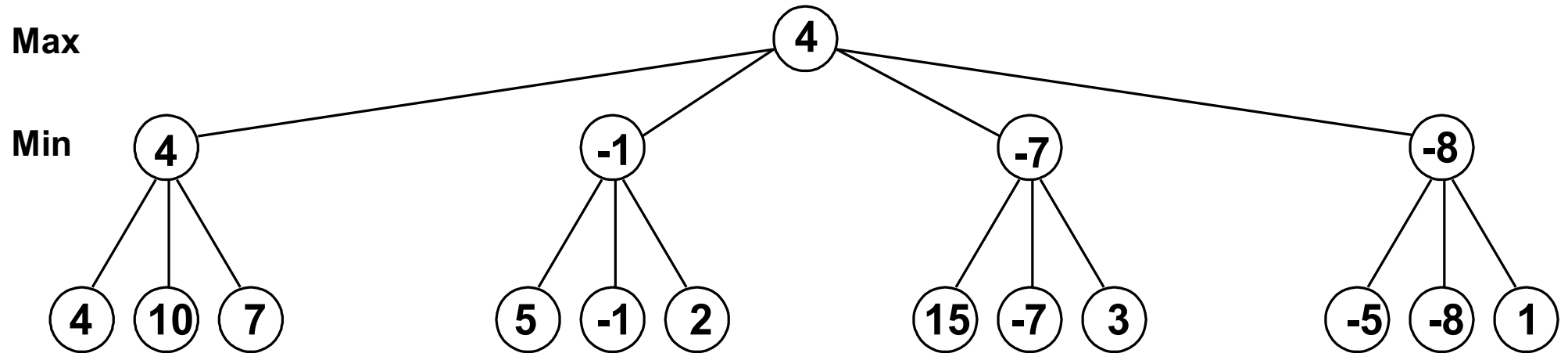
Ο Αλγόριθμος Alpha-Beta

- ❖ Ο Άλφα-Βήτα (Alpha-Beta - AB) αποφεύγει την αξιολόγηση καταστάσεων.
- ❖ Ο AB είναι όμοιος με τον Minimax, αλλά με κλάδεμα υποδένδρων.

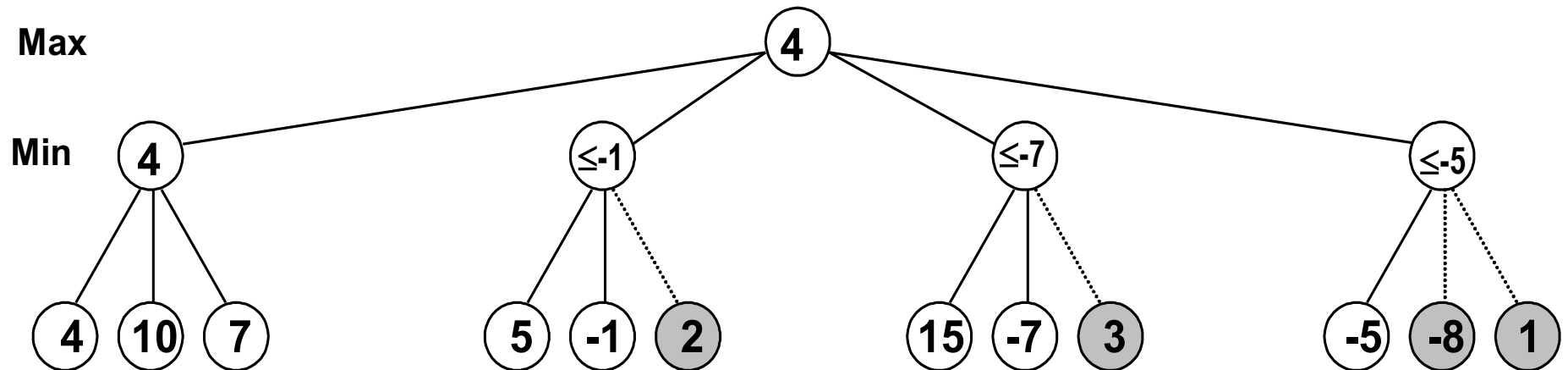


Σύγκριση του AB με τον Minimax (1/2)

Παράδειγμα Minimax



Παράδειγμα Alpha-Beta



Σύγκριση του AB με τον Minimax (2/2)

- ❖ Ο AB εξετάζει περίπου \sqrt{N} τερματικούς κόμβους, όπου N είναι οι τερματικοί κόμβοι που εξετάζει ο αλγόριθμος Minimax.
- ❖ Η απόδοσή του βελτιώνεται με διάφορες μεθόδους, όπως:
 - ❑ Ευριστικό κλάδεμα του δένδρου παιχνιδιού.
 - ❑ Δυναμική αντί στατικής συνάρτησης αξιολόγησης.
 - ❑ Αποθήκευση τιμών των τερματικών καταστάσεων (transposition tables).
 - ❑ Προκαθορισμένες κινήσεις (χωρίς αναζήτηση) σε αρχικές και τελικές φάσεις του παιχνιδιού (*Openings, End Game moves*).

Γενετικοί Αλγόριθμοι

- ❖ Σε αρκετές περιπτώσεις το μέγεθος ενός προβλήματος καθιστά απαγορευτική τη χρήση κλασικών μεθόδων αναζήτησης για την επίλυσή του.
- ❖ Στις περιπτώσεις αυτές βρίσκουν εφαρμογή πιθανοκρατικοί αλγόριθμοι οι οποίοι αν και δεν εγγυώνται ότι θα βρουν τη βέλτιστη λύση, είναι ικανοί να επιστρέψουν μια αρκετά καλή λύση σε εύλογο χρονικό διάστημα.
- ❖ Μια κατηγορία τέτοιων αλγορίθμων επίλυσης προβλημάτων είναι οι γενετικοί αλγόριθμοι (*genetic algorithms*), των οποίων ο βασικός μηχανισμός είναι εμπνευσμένος από τη Δαρβινική θεωρία της εξέλιξης (evolution) της φύσης.
- ❖ Οι γενετικοί αλγόριθμοι εκτελούν μία αναζήτηση στο χώρο των υποψηφίων λύσεων, με στόχο την εύρεση αποδεκτών, σύμφωνα με κάποιο κριτήριο, λύσεων.

Ιστορικά στοιχεία

- ❑ Το 1958 ο Friedberg, επιχείρησε να συνδυάσει μικρά προγράμματα FORTRAN, ωστόσο τα προγράμματα που προέκυψαν τις περισσότερες φορές δεν ήταν εκτελέσιμα.
- ❑ Το 1975 ο Holland έδωσε νέα ώθηση στο χώρο, χρησιμοποιώντας σειρές bits για να αναπαραστήσει λειτουργίες, με τρόπο τέτοιο ώστε κάθε συνδυασμός bits να είναι μια έγκυρη λειτουργία.

Θεωρία της εξέλιξης (evolution)

- ❖ Κανόνας της φυσικής επιλογής
 - ❑ Οι οργανισμοί που δε μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι πολλαπλασιάζονται μέσω της αναπαραγωγής.
 - ❑ Οι απόγονοι παρουσιάζουν μικρές διαφοροποιήσεις από τους προγόνους τους, ενώ συνήθως υπερισχύουν αυτοί που συγκεντρώνουν τα καλύτερα χαρακτηριστικά.
 - ❑ Σποραδικά συμβαίνουν τυχαίες μεταλλάξεις, από τις οποίες οι περισσότερες οδηγούν τα μεταλλαγμένα άτομα στο θάνατο, αν και είναι πιθανό, πολύ σπάνια όμως, να οδηγήσουν στη δημιουργία νέων "καλύτερων" οργανισμών.
 - ❑ Αν το περιβάλλον μεταβάλλεται με αργούς ρυθμούς, τα διάφορα είδη μπορούν να εξελίσσονται σταδιακά ώστε να προσαρμόζονται σε αυτό.

Γενική Μορφή Γενετικού Αλγόριθμου (1/2)

Βήματα:

- 1) Δημιούργησε τυχαία έναν αρχικό πληθυσμό Π , με N υποψήφια (μη αποδεκτές, δηλαδή. μη έγκυρες ή μη βέλτιστες, κλπ) λύσεις.
- 2) Βαθμολόγησε (δηλ. πόσο κοντά σε μια αποδεκτή λύση είναι) κάθε υποψήφια λύση χρησιμοποιώντας μια *συνάρτηση καταλληλότητας* (*fitness function*).
- 3) Σχημάτισε $N/2$ ζευγάρια όχι απαραίτητα μοναδικών γονέων, δίνοντας μεγαλύτερη προτεραιότητα στις πλέον κατάλληλες λύσεις.
- 4) Κάθε ζευγάρι *ζευγαρώνει* (mates), δίνοντας δύο νέες λύσεις, τους *απογόνους* (offsprings).
- 5) Ο νέος πληθυσμός Π' αποτελείται από το σύνολο των απογόνων και συνήθως αποτελεί βελτίωση του προηγούμενου πληθυσμού.
- 6) Η διαδικασία επαναλαμβάνεται για το νέο πληθυσμό Π'
 - ❖ Οι πιο συνηθισμένες συνθήκες τερματισμού της είναι η εύρεση μιας τέλειας λύσης με βάση τη συνάρτηση καταλληλότητας ή η σύγκλιση όλων των λύσεων σε μια.

Γενική Μορφή Γενετικού Αλγόριθμου (2/2)

1. Δημιούργησε έναν αρχικό πληθυσμό Π , με N υποψήφιας λύσεις.
2. Υπολόγισε την καταλληλότητα κάθε λύσης.
3. Όσο δεν ισχύει κάποια συνθήκη τερματισμού:
 - α. Επανάλαβε $N/2$ φορές τα ακόλουθα βήματα:
 - i. Επίλεξε δύο λύσεις από τον πληθυσμό Π .
 - ii. Συνδύασε τις δύο λύσεις για να βγάλεις δύο απογόνους.
 - iii. Υπολόγισε την καταλληλότητα των δύο απογόνων.
 - β. Δημιούργησε το νέο πληθυσμό Π' έχοντας υπόψη όλους τους νέους απογόνους που προέκυψαν από το βήμα 3α και θέσε $\Pi = \Pi'$

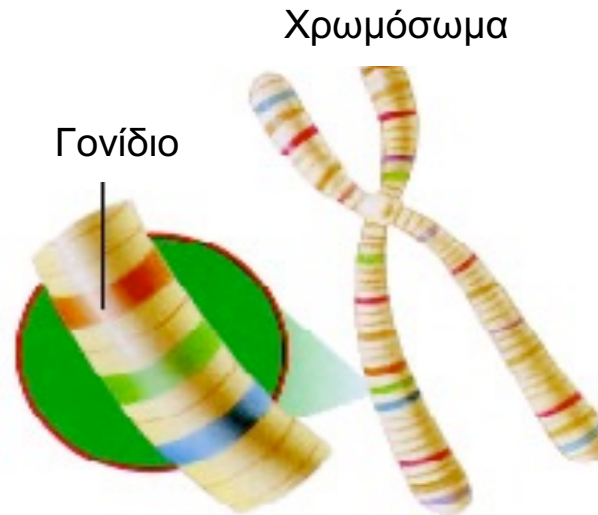
❖ Ένας γενετικός αλγόριθμος για ένα συγκεκριμένο πρόβλημα περιλαμβάνει πέντε συστατικά:

- Δημιουργία αρχικού πληθυσμού (συνήθως δημιουργείται τυχαία)
- Αναπαράσταση λύσεων
- Συνάρτηση καταλληλότητας
- Επιλογή γονέων

- Διαδικασία αναπαραγωγής

Αναπαράσταση Υποψήφιων Λύσεων (1/2)

- ❖ Στους βιολογικούς οργανισμούς, ένα χρωμόσωμα είναι ένα μεγάλο μόριο (ακολουθία) DNA και περιέχει έναν αριθμό γονιδίων.



- ❖ Στο πραγματικό DNA το αλφάβητο έχει μήκος τέσσερα και αποτελείται από τα γράμματα A, G, T και C που αντιστοιχούν στα τέσσερα διαφορετικά νουκλεοτίδια (βάσεις) που το συνθέτουν (Adenine, Guanine, Thymine και Cytosine).

Αναπαράσταση Υποψήφιας Λύσεων (2/2)

- ❖ Στην κλασική προσέγγιση των γενετικών αλγορίθμων, κάθε υποψήφια λύση αναπαρίσταται με μία συμβολοσειρά (string) ενός πεπερασμένου αλφάβητου.
 - ❑ Συνήθως χρησιμοποιείται το δυαδικό αλφάβητο, οπότε οι συμβολοσειρές ονομάζονται και δυαδικές συμβολοσειρές (bit-strings).
 - ❑ Στα περισσότερα προβλήματα οι λύσεις περιγράφονται με μεταβλητές διαφόρων τύπων δεδομένων, επομένως η διαδικασία της κωδικοποίησης περιλαμβάνει τη μετατροπή των τιμών αυτών των μεταβλητών στις αντίστοιχες δυαδικές.
- ❖ Κατ' αναλογία με τη βιολογία, η συμβολοσειρά συνήθως αναφέρεται και σαν χρωμόσωμα (*chromosome*) ενώ τα επιμέρους τμήματά της που κωδικοποιούν κάποιο χαρακτηριστικό, δηλαδή κάποια μεταβλητή, ονομάζονται γονίδια (*gene*).

Συνάρτηση Καταλληλότητας

- ❖ Δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό (συνήθως στο διάστημα $[0,1]$), που υποδηλώνει το πόσο κατάλληλο είναι.
- ❖ Η αξιολόγηση αυτή χρησιμοποιείται είτε από τη συνθήκη τερματισμού ή από τη διαδικασία της πιθανοκρατικής επιλογής τους για να συμπεριληφθούν (ή όχι) στον πληθυσμό της επόμενης γενιάς.

Διαδικασία Επιλογής Γονέων (1/4)

- ❖ Απόδοση πιθανοτήτων επιλογής προς αναπαραγωγή στα μέλη ενός πληθυσμού υποψηφίων λύσεων.
 - Κάποιοι γονείς με υψηλή τιμή στη συνάρτηση καταλληλότητας ενδέχεται να επιλεγούν προς αναπαραγωγή περισσότερες από μία φορές, ενώ κάποιοι γονείς με χαμηλή καταλληλότητα ενδέχεται να μην επιλεγούν καθόλου.

- ❖ Κατά τη διαδικασία επιλογής, αρχικά οι υποψήφιες λύσεις αντιγράφονται σε μια δεξαμενή ζευγαρώματος (*mating pool*).
 - Σε αυτήν αντιγράφονται μέλη του αρχικού πληθυσμού, με πιθανότητα ανάλογη της καταλληλότητάς τους.
 - Για την επιλογή των χρωμοσωμάτων που θα αντιγραφούν στη δεξαμενή ζευγαρώματος χρησιμοποιούνται αρκετές τεχνικές.

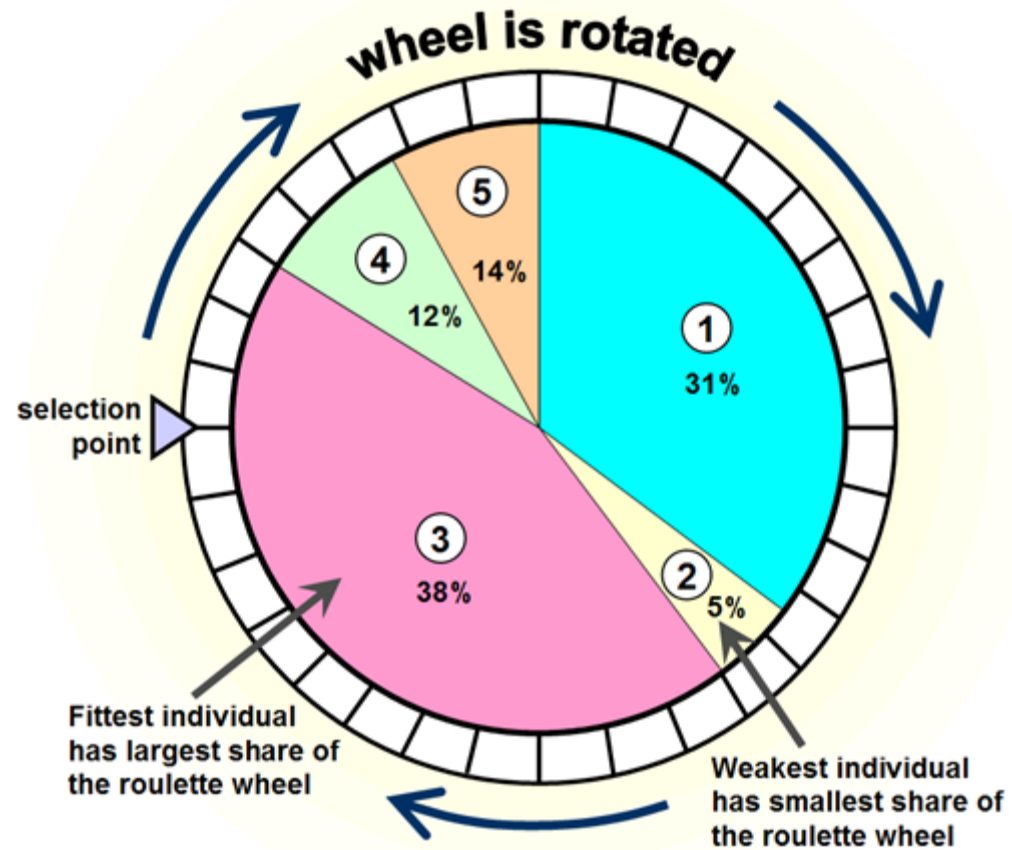
Διαδικασία Επιλογής Γονέων (2/4)

Τεχνική της αναλογικής καταλληλότητας

❖ (*fitness proportionate selection*)

□ Πιθανότητα επιλογής χρωμοσώματος x_i :

$$P(x_i) = \frac{\text{Καταλληλότητα}(x_i)}{\sum_{j=1}^N \text{Καταλληλότητα}(x_j)}$$



Διαδικασία Επιλογής Γονέων (3/4)

Τεχνική της ρουλέτας

1. Παράγεται το άθροισμα S όλων των τιμών αξιολόγησης των υποψηφίων λύσεων.
2. Επιλέγεται ένας τυχαίος αριθμός n , από το 0 μέχρι το S , χρησιμοποιώντας συνάρτηση ομοιόμορφης κατανομής για τη δημιουργία των τυχαίων αριθμών.
3. Επαναληπτικά εξετάζεται κάθε υποψήφια λύση και η τιμή της προστίθεται σε έναν καταχωρητή K .
4. Αν η τιμή του K γίνει μεγαλύτερη ή ίση του n , η τελευταία λύση επιλέγεται και ο K μηδενίζεται. Στην αντίθετη περίπτωση εκτελείται πάλι το 3.
5. Αν δεν έχει επιλεγεί ικανοποιητικός αριθμός υποψηφίων λύσεων εκτελείται το 2, αλλιώς ο αλγόριθμος τερματίζει.

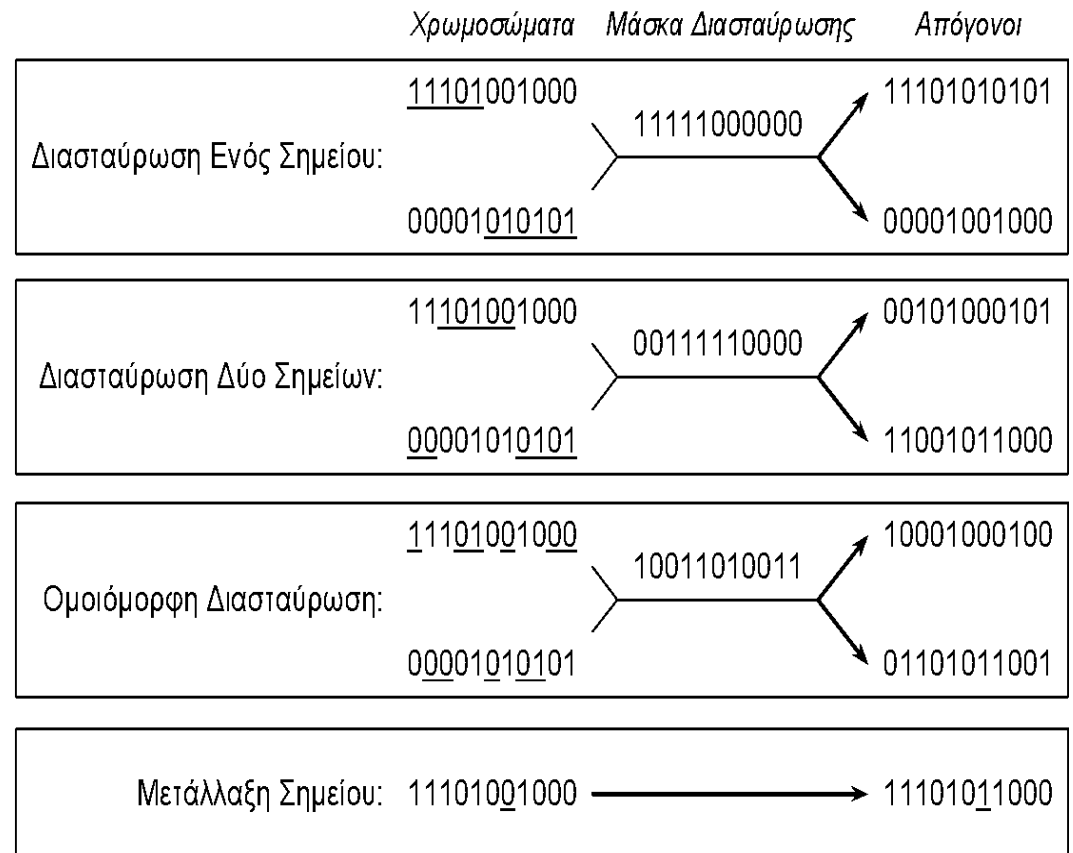
Διαδικασία Επιλογής Γονέων (4/4)

Τεχνική της επιλογής τουρνουά

- ❖ Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population.
 - choose k (the tournament size) individuals from the population at random
- ❖ The winner of each tournament is selected for crossover.
 - Deterministic tournament selection selects the best individual (when $p=1$) in any tournament.
 - Probabilistic tournament selection selects as follows:
 - choose the best individual from pool/tournament with a fixed probability p
 - choose the second best individual with probability $p*(1-p)$
 - choose the third best individual with probability $p*((1-p)^2)$
 - and so on...
- ❖ Tournament selection has several benefits: it is efficient to code, works on parallel architectures and allows the selection pressure to be easily adjusted.

Αναπαραγωγή

- ❖ Αναπαραγωγή είναι η διαδικασία δημιουργίας απογόνων.
- ❑ Σε αυτή εμπλέκονται ένα σύνολο από τελεστές οι οποίοι αντιστοιχούν σε διαδικασίες της βιολογικής εξέλιξης.
- ❖ Οι πιο συνηθισμένοι τελεστές είναι:
 - ❑ Διασταύρωση ενός σημείου (single-point crossover)
 - ❑ Διασταύρωση δύο σημείων (two-point crossover)
 - ❑ Ομοιόμορφη διασταύρωση (uniform crossover)
 - ❑ Μετάλλαξη σημείου (point mutation)



Σύγκλιση Πληθυσμού

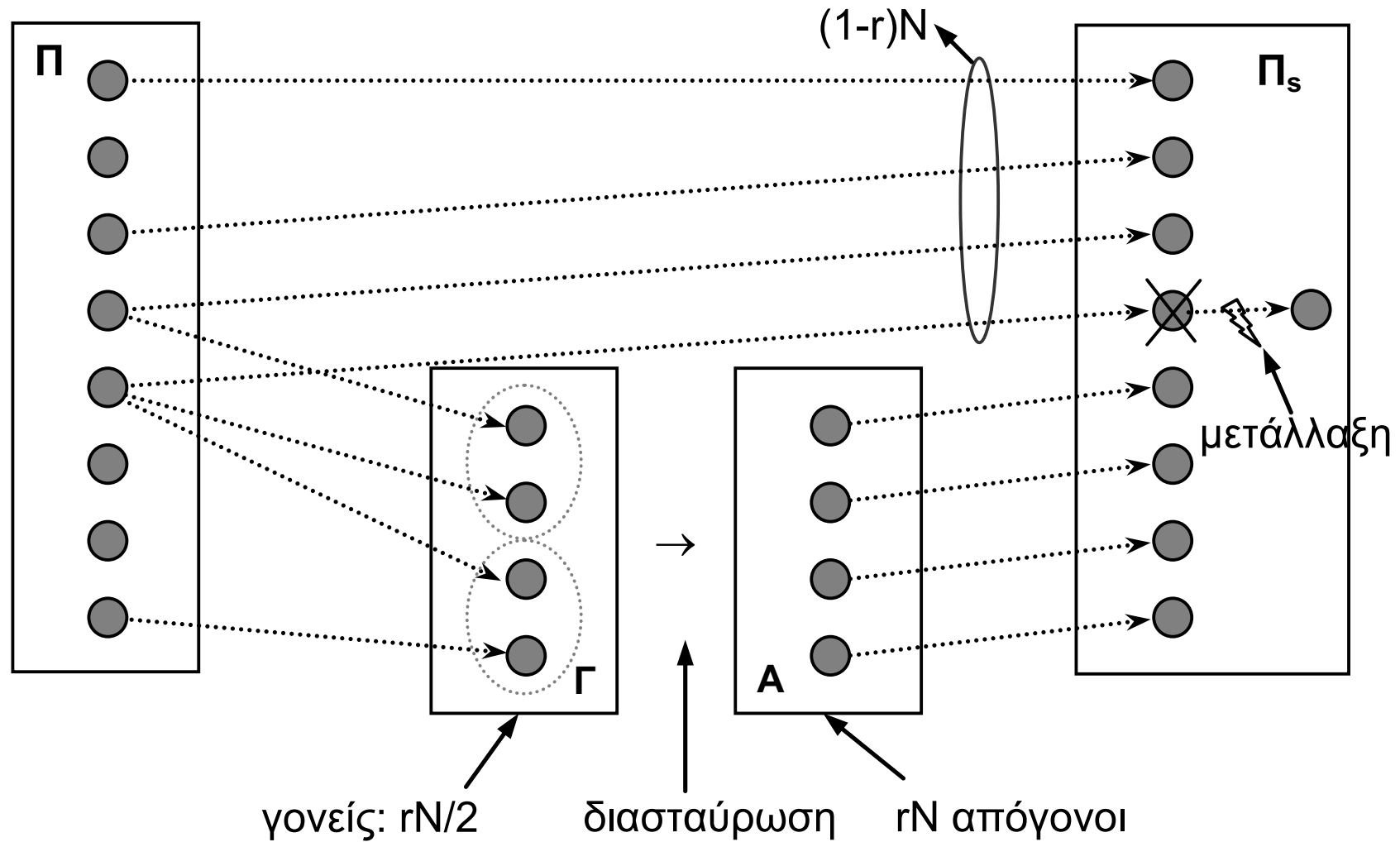
- ❖ **Σύγκλιση:** Η επικράτηση ενός χρωμοσώματος ή μικρών παραλλαγών του, σε μεγάλο ποσοστό στον πληθυσμό.
 - Με έναν αποδοτικό γενετικό αλγόριθμο, ο πληθυσμός θα πρέπει μετά από αρκετές επαναλήψεις να συγκλίνει προς το ολικό μέγιστο.
- ❖ Ένα γονίδιο συγκλίνει, όταν έχει την ίδια τιμή στο 95% των χρωμοσωμάτων.
- ❖ Ένας πληθυσμός συγκλίνει, όταν όλα τα γονίδιά του έχουν συγκλίνει.
- ❖ Εναλλακτική προσέγγιση στο θέμα της εξέλιξης του πληθυσμού είναι η μη ανανέωση ολόκληρου του πληθυσμού σε κάθε γενιά, αλλά ενός μέρους αυτού.

Μερική Ανανέωση Πληθυσμού

- ❖ Η μέθοδος της μερικής ανανέωσης προσεγγίζει πιο πολύ στην πραγματικότητα, αφού εκεί συνυπάρχουν πάντα σε κάποιο βαθμό οι διαφορετικές γενεές.
- ❖ Μάλιστα δίνεται η δυνατότητα στους απογόνους να ανταγωνιστούν τους γονείς τους, επικρατώντας και πάλι ο καλύτερος.
- ❖ Υπάρχουν δύο θέματα στην προσέγγιση της μερικής ανανέωσης: πώς θα επιλεγούν οι γονείς που θα ζευγαρώσουν και πώς θα επιλεγεί ισάριθμος αριθμός γονέων που θα αποχωρήσουν, ώστε να αφήσουν χώρο για τους απογόνους.

Γενικός γενετικός αλγόριθμος

Σχηματική λειτουργία



Γενικός γενετικός αλγόριθμος

Περιγραφή (1/2)

- ❖ Στον αλγόριθμο αυτόν, σε κάθε επανάληψη παράγεται μια νέα γενιά χρωμοσωμάτων Ps που προκύπτει μέσω δεδομένης διαδικασίας από τον αρχικό πληθυσμό P .
- ❖ Αρχικός και τελικός πληθυσμός έχουν μέγεθος N αλλά ποσοστό r του αρχικού πληθυσμού αντικαθίσταται σε κάθε κύκλο.
- ❖ Πρώτα επιλέγεται ένας συγκεκριμένος αριθμός χρωμοσωμάτων του υπάρχοντα πληθυσμού για να συμπεριληφθεί απευθείας στην επόμενη γενιά (μερική ανανέωση).
- ❖ Στη συνέχεια, παράγονται επιπλέον μέλη μέσω μιας διαδικασίας διασταύρωσης.
- ❖ Τόσο η επιλογή των μελών της υπάρχουσας γενιάς που θα συνεχίσουν και στην επόμενη όσο και των χρωμοσωμάτων-γονέων γίνεται πιθανοκρατικά.
- ❖ Στο σημείο αυτό επιλέγεται τυχαία ένα ποσοστό m του πληθυσμού αυτού στο οποίο θα συμβούν τυχαίες μεταλλάξεις.
- ❖ Στο Σχήμα, Γ είναι η δεξαμενή ζευγαρώματος από όπου, με κάποια διαδικασία διασταύρωσης, προκύπτει το σύνολο A των απογόνων.

Γενικός γενετικός αλγόριθμος

Περιγραφή (2/2)

- ❖ Άρα σε ένα γενετικό αλγόριθμο πρέπει να ορισθούν οι ακόλουθες παράμετροι:
 - ❑ Καταλληλότητα: η συνάρτηση καταλληλότητας.
 - ❑ Όριο_Καταλληλότητας: η ελάχιστη τιμή καταλληλότητας που πρέπει να επιτευχθεί από ένα χρωμόσωμα, ώστε να τερματίσει η διαδικασία.
 - ❑ N: ο αριθμός των χρωμοσωμάτων του πληθυσμού.
 - ❑ r: το ποσοστό του πληθυσμού που αντικαθίσταται σε κάθε κύκλο.
 - ❑ m: το ποσοστό μετάλλαξης.
- ❖ Τα βήματα του αλγορίθμου δίνονται στη συνέχεια:

Γενικός γενετικός αλγόριθμος

Γενετικός_Αλγόριθμος (Καταλληλότητα, Όριο_Καταλληλότητας, N , r , m)

1. Αρχικοποίηση του Πληθυσμού: $\Pi \leftarrow$ Τυχαία παραγωγή N χρωμοσωμάτων
2. Αξιολόγηση: $\forall x \in \Pi$ υπολόγισε την ποσότητα Καταλληλότητα(x)
3. Επανάλαβε: Όσο $\max(\text{Καταλληλότητα}(x), \forall x \in \Pi) < \text{Όριο_Καταλληλότητας}$
 - i. Επέλεξε πιθανοκρατικά $(1-r)N$ μέλη του Π και πρόσθεσέ τα στο Π_s .

Η πιθανότητα $P(x_i)$ να επιλεγεί ένα χρωμόσωμα x από το Π δίνεται από τη σχέση:

$$P(x_i) = \frac{\text{Καταλληλότητα}(x_i)}{\sum_{j=1}^N \text{Καταλληλότητα}(x_j)}$$

ii. Διασταύρωση: Πιθανοκρατικά επέλεξε $rN/2$ ζεύγη χρωμοσωμάτων από το Π σύμφωνα με την πιθανότητα $P(x_i)$.

\forall ζεύγος (x_1, x_2) , πάραξε δύο απογόνους με εφαρμογή του τελεστή διασταύρωσης. Πρόσθεσε τους απογόνους στο Π_s .

iii. Μετάλλαξη: Επέλεξε $m\%$ από τα μέλη του Π_s με ομοιόμορφη πιθανότητα. Για κάθε ένα, αντέστρεψε ένα τυχαία

επιλεγμένο bit.

iv. Ενημέρωση: $\Pi \leftarrow \Pi_s$.

v. $\forall x \in \Pi$ υπολόγισε την ποσότητα Καταλληλότητα(x)

4. Επέστρεψε το χρωμόσωμα με τη μεγαλύτερη Καταλληλότητα

Παράδειγμα (1/2)

- ❖ Έστω ένας πληθυσμός αποτελείται από 4 χρωμοσώματα.
- ❖ Στην 3^η στήλη φαίνονται οι τιμές της συνάρτησης καταλληλότητας, ενώ στην 4^η οι πιθανότητες επιλογής προς αναπαραγωγή κάθε χρωμοσώματος, ανηγμένες στο διάστημα $[0, 1]$.
- Οι τιμές της 4^{ης} στήλης προκύπτουν από το λόγο της καταλληλότητας του χρωμοσώματος προς το άθροισμα των καταλληλοτήτων όλων των χρωμοσωμάτων του πληθυσμού.

Υποψήφιες λύσεις	bit-string	Καταλληλότητα	Πιθανότητα επιλογής
A	000110010111	8	0.32
B	111010101100	6	0.24
Γ	001110101001	6	0.24
Δ	111011011100	5	0.20

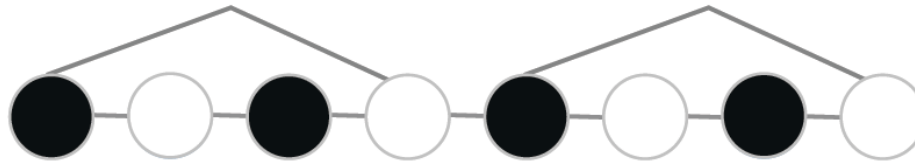
- ❖ Έστω ότι το αποτέλεσμα της διαδικασίας επιλογής είναι το $[A, B, B, \Gamma]$ και
- τα ζευγάρια που σχηματίζονται τυχαία είναι τα (B, A) και (B, Γ) και
- Ως τελεστής αναπαραγωγής του πρώτου ζευγαριού επιλέγεται η διασταύρωση ενός σημείου με μάσκα διασταύρωσης "111100000000" ενώ το για το δεύτερο ο ίδιος τελεστής με μάσκα διασταύρωσης "11111111000"

Παράδειγμα (2/2)

Γονέας 1	(B) 1110-10101100	(B) 1110-10101100	(B) 111010101-100	(B) 111010101-100
Γονέας 2	(A) 0001-10010111	(A) 0001-10010111	(Γ) 001110101-001	(Γ) 001110101-001
Απόγονος	1 ^{ος}	2 ^{ος}	3 ^{ος}	4 ^{ος}
Διασταύρωση	<i>1110-10010111</i>	<i>0001-10101100</i>	<i>111010101-001</i>	<i>001110101-100</i>
Μετάλλαξη	<i>1110-10010111</i>	<i>0001-10101100</i>	<i>111<u>1</u>0101-001</i>	<i>001110101-10<u>1</u></i>

Το πρόβλημα χρωματισμού γράφων

- ❖ Βρείτε τα χρώματα των κόμβων του γράφου (άσπρο ή μαύρο) έτσι ώστε κόμβοι που ενώνονται να έχουν διαφορετικά χρώματα



- ❖ Αναπαράσταση
 - διανύσματα δυαδικών αριθμών μήκους ίσου με το πλήθος των κόμβων (0 για άσπρο, 1 για μαύρο)
 - για παράδειγμα, το χρωμόσωμα του σχήματος θα έχει τη μορφή $v = \langle 1, 0, 1, 0, 1, 0, 1, 0 \rangle$
- ❖ Συνάρτηση καταλληλότητας
 - Αριθμός «προβληματικών» ενώσεων
- ❖ Επιλογή Γονέων
 - Π.χ. με τεχνική της ρουλέτας
- ❖ Αναπαραγωγή
 - Π.χ. Διασταύρωση ενός σημείου ή μόνο με μετάλλαξη ενός bit

Το πρόβλημα του πλανόδιου πωλητή

- ❖ Εύρεση της σειράς με την οποία ένας πωλητής πρέπει να περάσει από όλες τις πόλεις ενός συνόλου πόλεων και να επιστρέψει στην αρχική, ώστε να έχει το μικρότερο δυνατό κόστος, όπως αυτό εκφράζεται κάθε φορά (π.χ. χρόνος, χρήματα κλπ).
- ❖ Αναπαράσταση με διανύσματα ακεραίων αριθμών μήκους ίσου με το πλήθος των πόλεων. Έτσι, για παράδειγμα, ένα χρωμόσωμα θα έχει τη μορφή $v = \langle i_1 i_2 \dots i_n \rangle$, όπου $i_1, i_2, \dots, i_n \in 1..n$ και $i_j \neq i_k$ για $j \neq k$.
- ❖ Δημιουργία αρχικού πληθυσμού με τυχαίο τρόπο.
- ❖ Απλή συνάρτηση καταλληλότητας (π.χ. το αντίστροφο του κόστους της λύσης)
- ❖ Αναπαραγωγή: Μπορεί να προκύψουν μη-έγκυρα χρωμοσώματα !
 - ❑ Έλεγχος των απογόνων και επιδιόρθωση των μη-έγκυρων χρωμοσωμάτων.
 - ❑ Τροποποιημένες τεχνικές διασταύρωσης και μετάλλαξης, οι οποίες δίνουν πάντα έγκυρα χρωμοσώματα.
 - ❑ $P1 = 1\ 2\ 3\ | 4\ 5\ 6\ 7\ | 8\ 9$ $P2 = 4\ 5\ 2\ | 1\ 8\ 7\ 6\ | 9\ 3$
 - ❑ Για τη συμπλήρωση του απογόνου O1 θα ληφθούν υπόψη τα στοιχεία του P2 με τη σειρά 9 3 4 5 2 1 8 7 6
 - ❑ $O1 = 2\ 1\ 8\ | 4\ 5\ 6\ 7\ | 9\ 3$ $O2 = 3\ 4\ 5\ | 1\ 8\ 7\ 6\ | 9\ 2$

Εφαρμογές (1/2)

❖ Εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων.

□ Η εύρεση του μέγιστου μιας συνάρτησης δεν είναι καθόλου εύκολη υπόθεση για συναρτήσεις πολλών μεταβλητών, οι οποίες εμφανίζουν ασυνέχειες, θόρυβο, κλπ.

□ Το πλεονέκτημα που εμφανίζει η εφαρμογή τους σε αυτά τα προβλήματα είναι ότι η συνάρτηση καταλληλότητας είναι δεδομένη.

❖ Επεξεργασία εικόνων

□ Αναγνώριση προτύπων, όπως ακμές, επιφάνειες, ακόμη και αντικείμενα, σε ψηφιοποιημένες εικόνες.

❖ Συνδυαστική βελτιστοποίηση.

□ Το κλασικό πρόβλημα κατανομής πόρων σε δραστηριότητες, με σκοπό τη μεγιστοποίηση του οφέλους ή την ελάττωση του κόστους.

□ Ο έλεγχος όλων των υποψήφιων λύσεων να είναι αδύνατος (συνδυαστική έκρηξη)

□ Γνωστά προβλήματα αυτής της κατηγορίας: του *πλανόδιου πωλητή*, η *αποθήκευση κιβωτίων*, *σχεδίαση VLSI κυκλωμάτων*, *καταμερισμός εργασιών*, *ωρολόγιο πρόγραμμα*

Εφαρμογές (2/2)

❖ Σχεδίαση

- ❑ Κατασκευών και εξαρτημάτων, με ζητούμενο τόσο την εύρεση μιας λύσης, όσο και τη βελτιστοποίησή της.
- ❑ Οι αλγόριθμοι μπορούν να δοκιμάσουν συνδυασμούς και ιδέες που ο ανθρώπινος νους δε θα δοκίμαζε ποτέ, δίνοντας ενίοτε πρωτότυπα αποτελέσματα.

❖ Μηχανική μάθηση

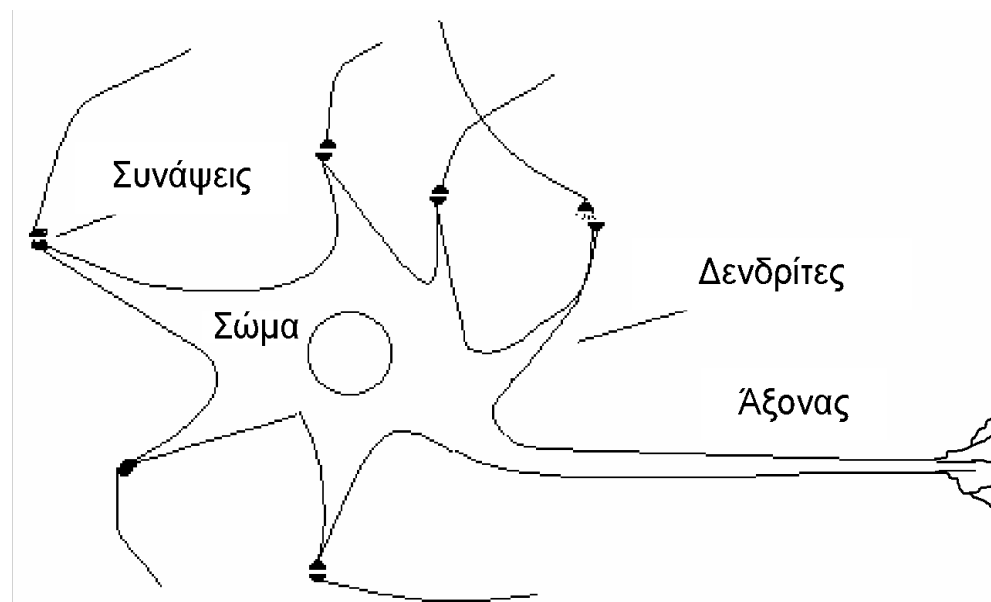
- ❑ Στα συστήματα μηχανικής μάθησης οι γενετικοί αλγόριθμοι μπορεί να χρησιμοποιηθούν για την προσέγγιση συναρτήσεων.
- ❑ Η πιο γνωστή εφαρμογή είναι αυτή των συστημάτων ταξινόμησης (classifier systems), ωστόσο οι γενετικοί αλγόριθμοι έχουν χρησιμοποιηθεί και σε παιχνίδια, επίλυση λαβυρίνθων, καθώς και για πολιτικές και οικονομικές αναλύσεις.

Νευρωνικά Δίκτυα (ΝΔ) - Εισαγωγή

- ❖ Είναι μια ιδιαίτερη προσέγγιση στη δημιουργία συστημάτων με νοημοσύνη.
- ❑ δεν αναπαριστούν ρητά τη γνώση
- ❑ δεν υιοθετούν ειδικά σχεδιασμένους αλγόριθμους αναζήτησης.
- ❖ Βασίζονται σε βιολογικά πρότυπα (ανθρώπινο εγκέφαλο)

Βιολογικός Νευρώνας

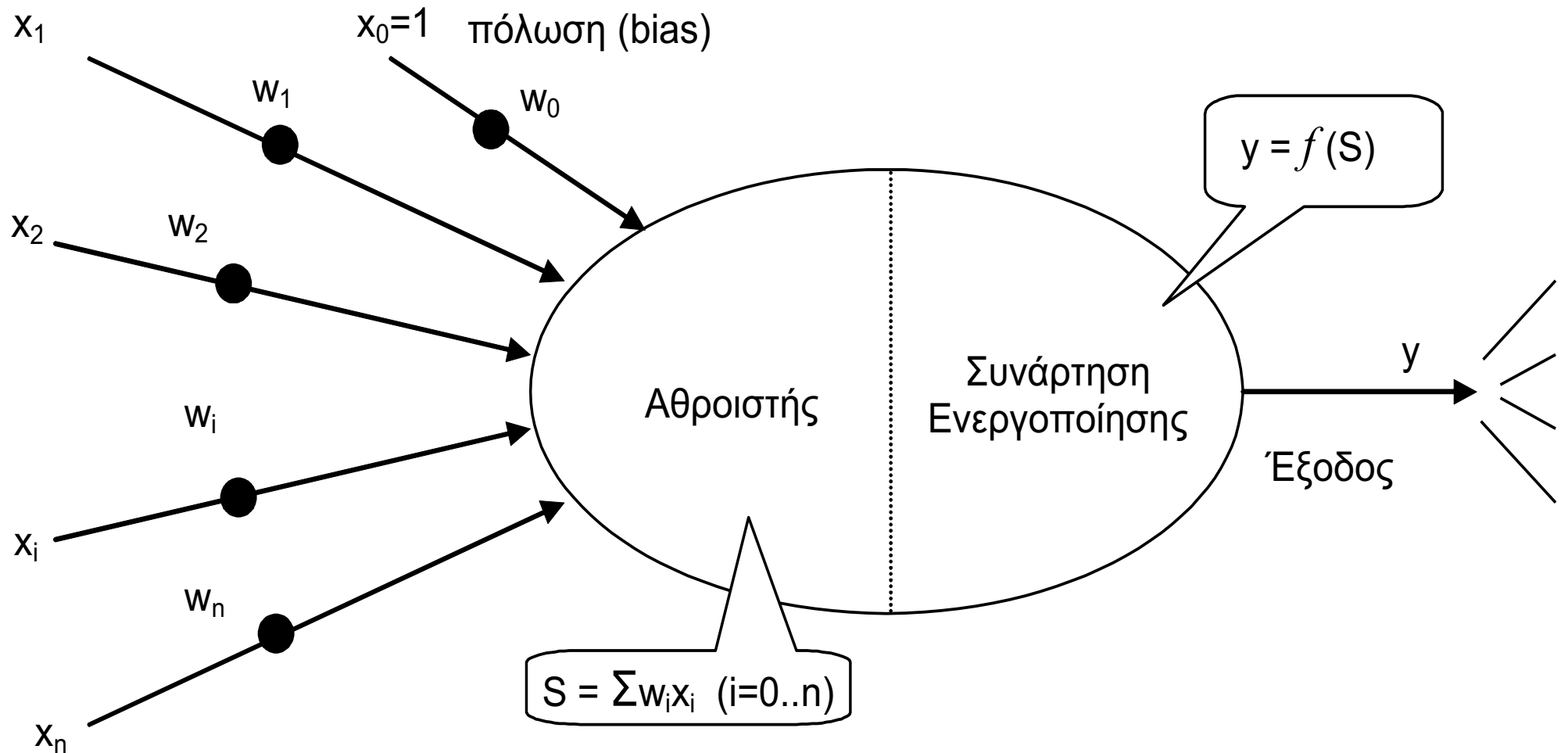
- ❖ μάθηση και μνήμη: μεταβολή στην αγωγιμότητα των συνάψεων
- ❖ Τα σήματα που εισέρχονται στο σώμα μέσω των δενδριτών, συνδυάζονται και αν το αποτέλεσμα ξεπερνά κάποιο κατώφλι, διαδίδεται μέσω του άξονα προς άλλους νευρώνες.



Φυσικά Νευρωνικά Δίκτυα

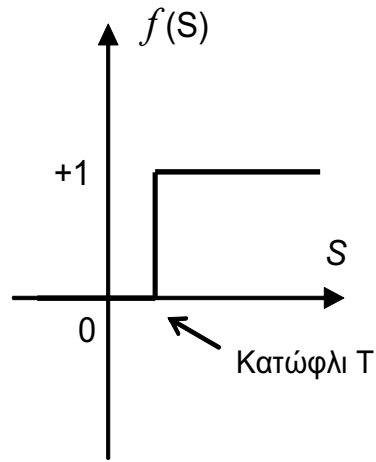
- ❖ ανθρώπινος εγκέφαλος:
 - ❑ ~100 δισεκατομμύρια νευρώνες .
 - ❑ κάθε νευρώνας συνδέεται κατά μέσο όρο με 1000 άλλους νευρώνες
 - ❑ ~100 τρισεκατομμύρια συνάψεις
- ❖ η αντιγραφή είναι εφικτή μόνο περιορισμένη κλίμακα
- ❖ χρόνος απόκρισης των βιολογικών νευρώνων: της τάξης msec
 - ❑ ...αλλά, λαμβάνει πολύπλοκες αποφάσεις, εκπληκτικά γρήγορα.
- ❖ η υπολογιστική ικανότητα του εγκεφάλου και η πληροφορία που περιέχει είναι διαμοιρασμένα σε όλο του τον όγκο
 - ❑ παράλληλο και κατανεμημένο υπολογιστικό σύστημα.

Μοντέλο Τεχνητού Νευρώνα (*artificial neuron*)

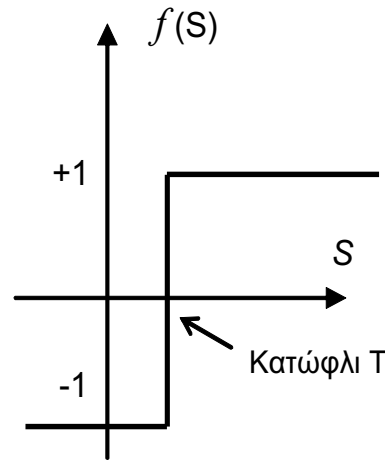


Συναρτήσεις Ενεργοποίησης

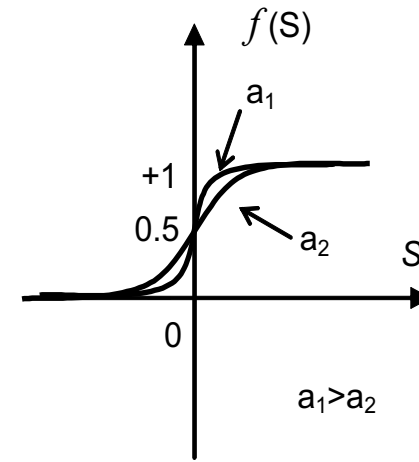
❖ Βασική απαίτηση: να είναι μη γραμμική ώστε να μπορεί να μοντελοποιεί μη γραμμικά φαινόμενα.



α) Βηματική Συνάρτηση



β) Συνάρτηση Προσήμου



γ) Λογιστική Συνάρτηση

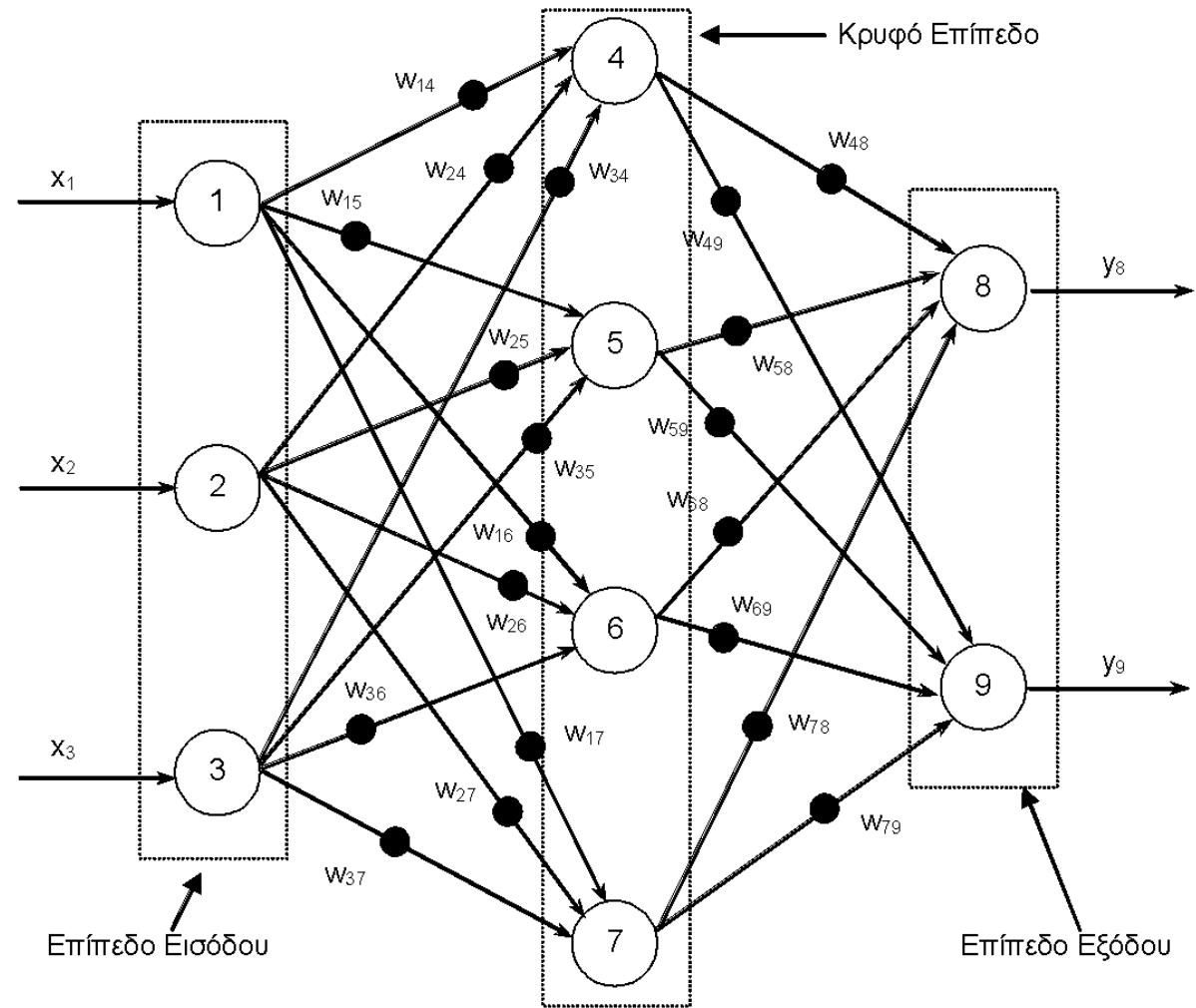
❖ Η λογιστική (*logistic*) συνάρτηση - μέλος οικογένειας σιγμοειδών συναρτήσεων.

$$\Phi(S) = \frac{1}{1 + e^{-a \cdot S}}$$

Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

❖ Συστήματα επεξεργασίας δεδομένων που αποτελούνται από ένα πλήθος τεχνητών νευρώνων οργανωμένων σε δομές παρόμοιες με αυτές του ανθρώπινου εγκεφάλου.

❖ Συντομογραφία για πολυεπίπεδα ΤΝΔ: $(p, m_1, m_2, \dots, m_q, n)$



Χαρακτηριστικά – Ορολογία

❖ Οι νευρώνες των διαφόρων στρωμάτων μπορεί να είναι:

- Πλήρως συνδεδεμένοι (*fully connected*)
- Μερικά συνδεδεμένοι (*partially connected*)

❖ Τρόφοδοτες ρίζονται ως:

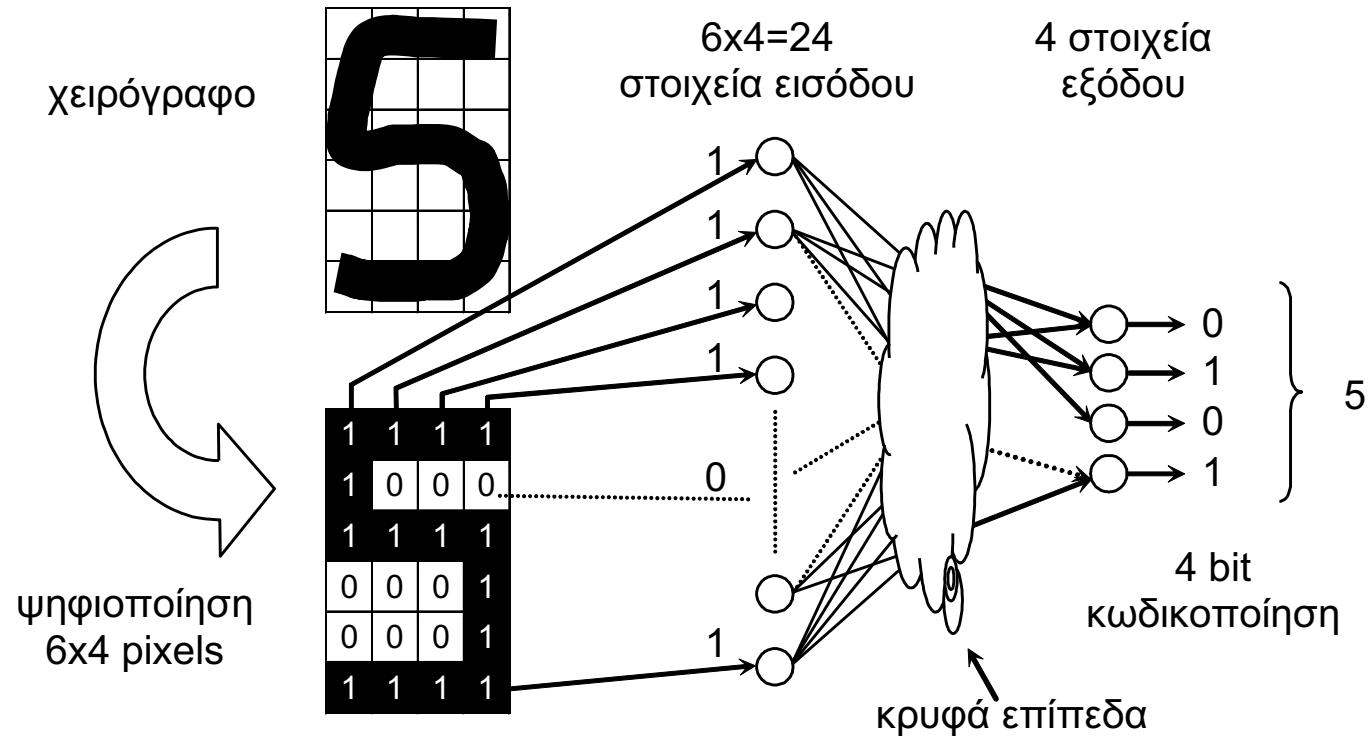
- Δίκτυα απλής τροφοδότησης (*feedforward*)

- Δίκτυα με ανατροφοδότηση (*feedback ή recurrent*)

❖ Στην πλειοψηφία των εφαρμογών χρησιμοποιούνται δίκτυα απλής τροφοδότησης.

Μάθηση και Ανάκληση

- ❖ *Μάθηση - learning* (ή *εκπαίδευση - training*) είναι η διαδικασία της τροποποίησης της τιμής των βαρών του δικτύου, ώστε δοθέντος συγκεκριμένου διανύσματος εισόδου να παραχθεί συγκεκριμένο διάνυσμα εξόδου.
- ❖ *Ανάκληση (recall)* είναι η διαδικασία του υπολογισμού ενός διανύσματος εξόδου για συγκεκριμένο διάνυσμα εισόδου και τιμές βαρών.
- ❖



3 Είδη Μάθησης

□ Μάθηση υπό Επίβλεψη (*supervised learning*)

Δίνονται ζευγάρια διανυσμάτων εισόδου – επιθυμητής εξόδου και το δίκτυο παράγει με τα τρέχοντα βάρη μία έξοδο

Υπολογίζεται το σφάλμα = έξοδος – επιθυμητή έξοδος

Γίνεται αναπροσαρμογή βαρών

□ Μάθηση χωρίς Επίβλεψη (*unsupervised learning*)

Η απόκριση του δικτύου βασίζεται στην ικανότητά του να αυτο-οργανώνεται με βάση τις εισόδους (δεν υπάρχουν επιθυμητές έξοδοι)

Αυτή γίνεται με στόχο για συγκεκριμένο σύνολο εισόδων να αντιδρά ισχυρά ένας συγκεκριμένος νευρώνας

Κατηγοριοποίηση δεδομένων εισόδου

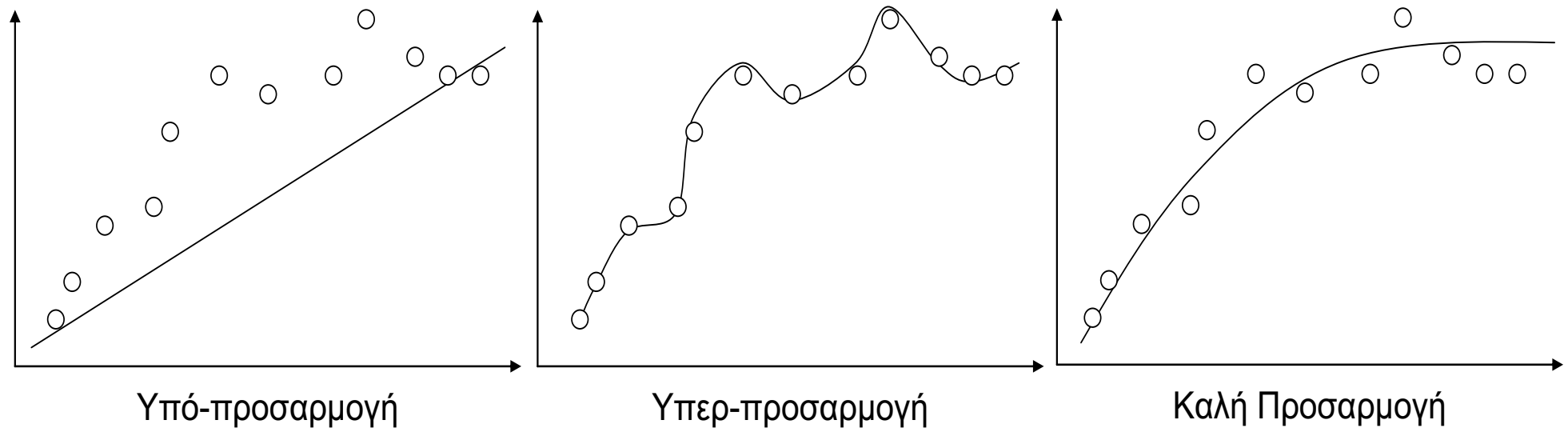
□ Βαθμολογημένη Μάθηση (*reinforcement learning*)

Η έξοδος χαρακτηρίζεται ως καλή ή κακή και τα βάρη αναπροσαρμόζονται αναλόγως

Στην πράξη, στις περισσότερες εφαρμογές ΤΝΔ χρησιμοποιείται μάθηση υπό επίβλεψη, για την οποία υπάρχουν αρκετοί αλγόριθμοι.

Χαρακτηριστικά Εκπαίδευσης

- ❖ υποπροσαρμογής ή ατελούς μάθησης (*underfitting*)
- ❖ υπερπροσαρμογής (*overfitting*)



Δεδομένα Εκπαίδευσης

- ❖ χρήση σε κύκλους εκπαίδευσης που ονομάζονται εποχές (*epochs*)
 - ❑ μάθηση δέσμης (*batch learning*)
 - ❑ επαυξητική μάθηση (*incremental learning*)
 - ❑ συνδυασμός των δύο παραπάνω μεθόδων

- ❖ Η εκπαίδευση τερματίζεται όταν το κριτήριο ελέγχου της ποιότητας του δικτύου φτάσει σε κάποια επιθυμητή τιμή.

- ❖ Κριτήρια Ελέγχου Ποιότητας
 - ❑ μέσο σφάλμα του συνόλου εκπαίδευσης
 - ❑ μεταβολή του μέσου σφάλματος του συνόλου εκπαίδευσης

- ❖ Κανονικοποίηση δεδομένων εκπαίδευσης και ελέγχου
 - ❑ (τα δεύτερα, με βάση τις παραμέτρους κανονικοποίησης των πρώτων).

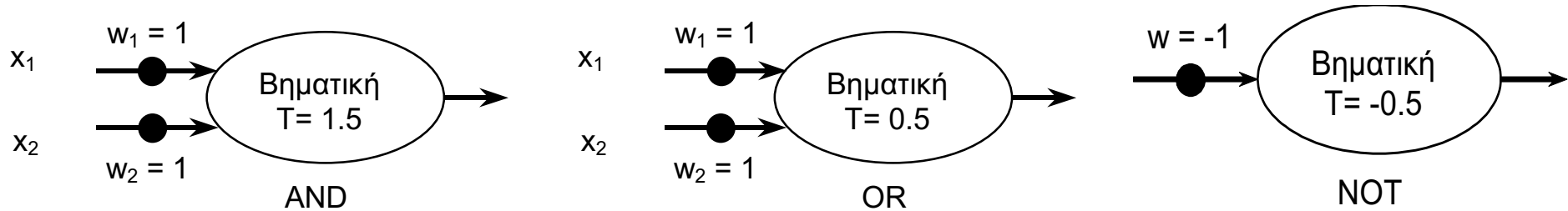
Βασικές Ιδιότητες των ΤΝΔ

- ❖ Ικανότητα να μαθαίνουν μέσω παραδειγμάτων (learn by example).
- ❖ Η δυνατότητα θεώρησής τους ως κατανεμημένη μνήμη (distributed memory) και ως μνήμη συσχέτισης (associative memory).
- ❖ Η μεγάλη τους ανοχή σε σφάλματα (fault-tolerant).
- ❖ Η εξαιρετική ικανότητά τους για αναγνώριση προτύπων (pattern recognition).

Perceptron

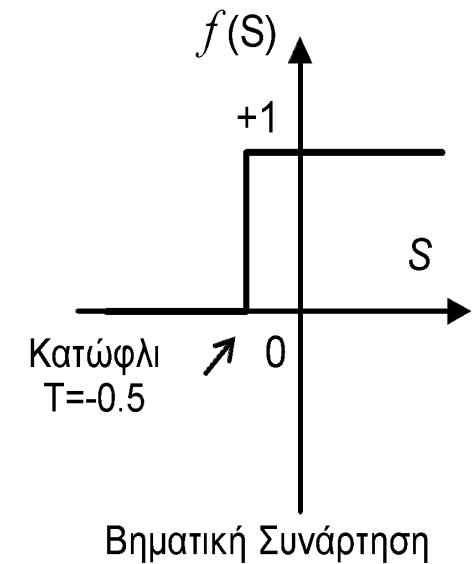
- ❖ Η πιο απλή τοπολογία δικτύου με απλή τροφοδότηση.
- ένας νευρώνας, βηματική συνάρτηση, μάθηση με επίβλεψη

Υλοποίηση Λογικών Συναρτήσεων με Perceptron



❖ **Παράδειγμα:** υλοποίηση του NOT:

- ❑ βηματική συνάρτηση ενεργοποίησης με κατώφλι $T = -0.5$
- ❑ για $X = 0$ ή $X = 1$
- ❑ αντιστρέφει την έξοδο

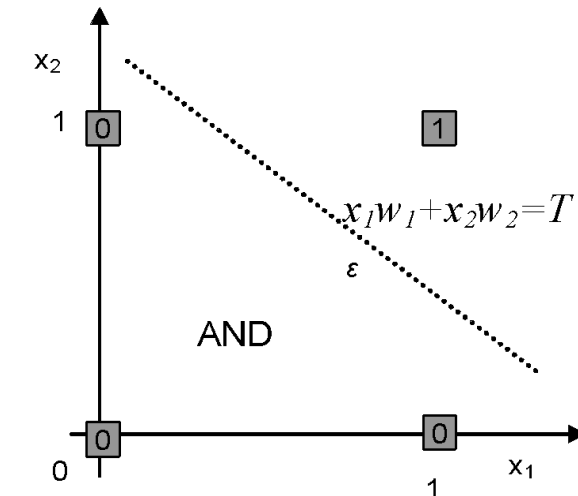


Γραμμική Διαχωρισιμότητα (1/3)

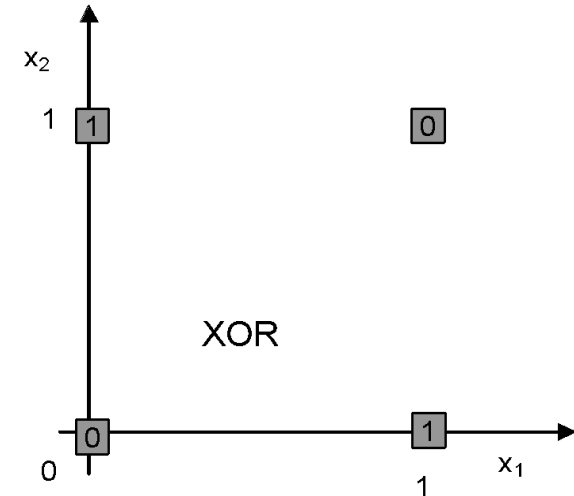
❖ Ένα perceptron με n γραμμές εισόδου μπορεί να θεωρηθεί ότι αναπαριστά ένα υπερεπίπεδο $n-1$ διαστάσεων που διαχωρίζει τα διανύσματα εισόδου σε δύο ομάδες, ανάλογα με την έξοδο.

❖ γραμμικώς διαχωρίσιμα προβλήματα (*linearly separable*).

❖ Παραδείγματα:



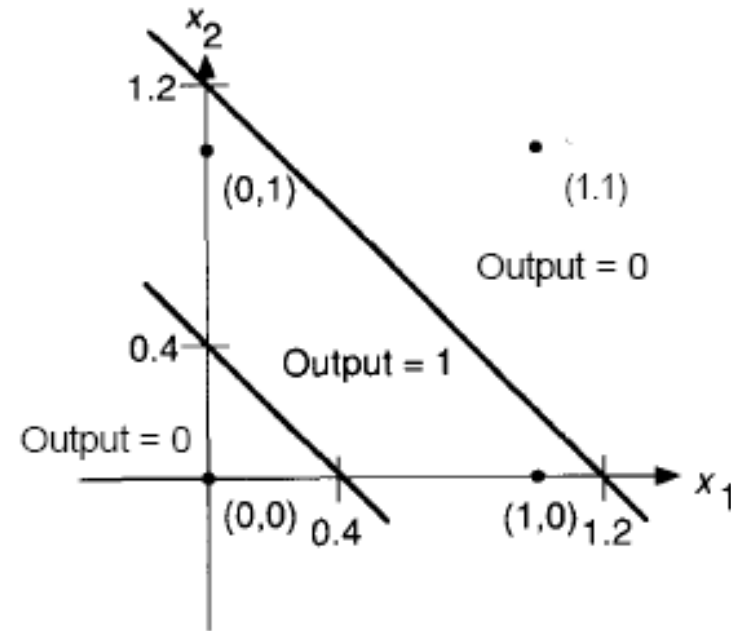
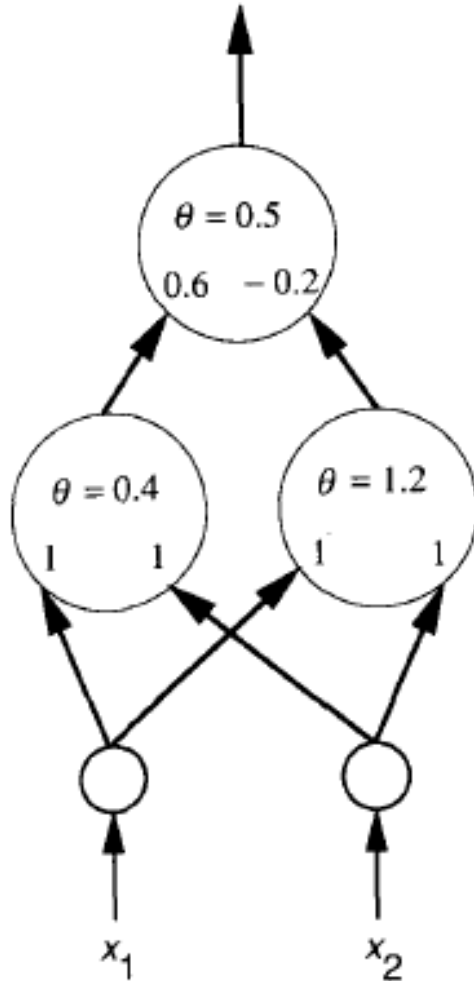
Συνάρτηση AND
(γραμμικώς διαχωρίσιμη)



Συνάρτηση XOR
(μη γραμμικώς διαχωρίσιμη)

❖ Τα μη γραμμικώς διαχωρίσιμα προβλήματα απαιτούν τη χρήση ΤΝΔ με ενδιάμεσα κρυφά επίπεδα

Γραμμική Διαχωριστικότητα (2/3)



Γραμμική Διαχωρισιμότητα (3/3)

- ❖ Γιατί τα ΤΝΔ με κρυφά επίπεδα (και νευρώνες) έχουν καλύτερη διαχωρισιμότητα;
 - Ο χώρος των τιμών εισόδου οριοθετείται με πιο πολύπλοκο τρόπο εξαιτίας της πολυπλοκότητας των συνδέσεων

Μάθηση σε Perceptron

❖ αλγόριθμος μεταβολής βαρών

Μέχρις ότου ικανοποιηθεί η συνθήκη τερματισμού της εκπαίδευσης επανάλαβε:
Για κάθε ζευγάρι εισόδου x και επιθυμητής εξόδου t από το σύνολο εκπαίδευσης

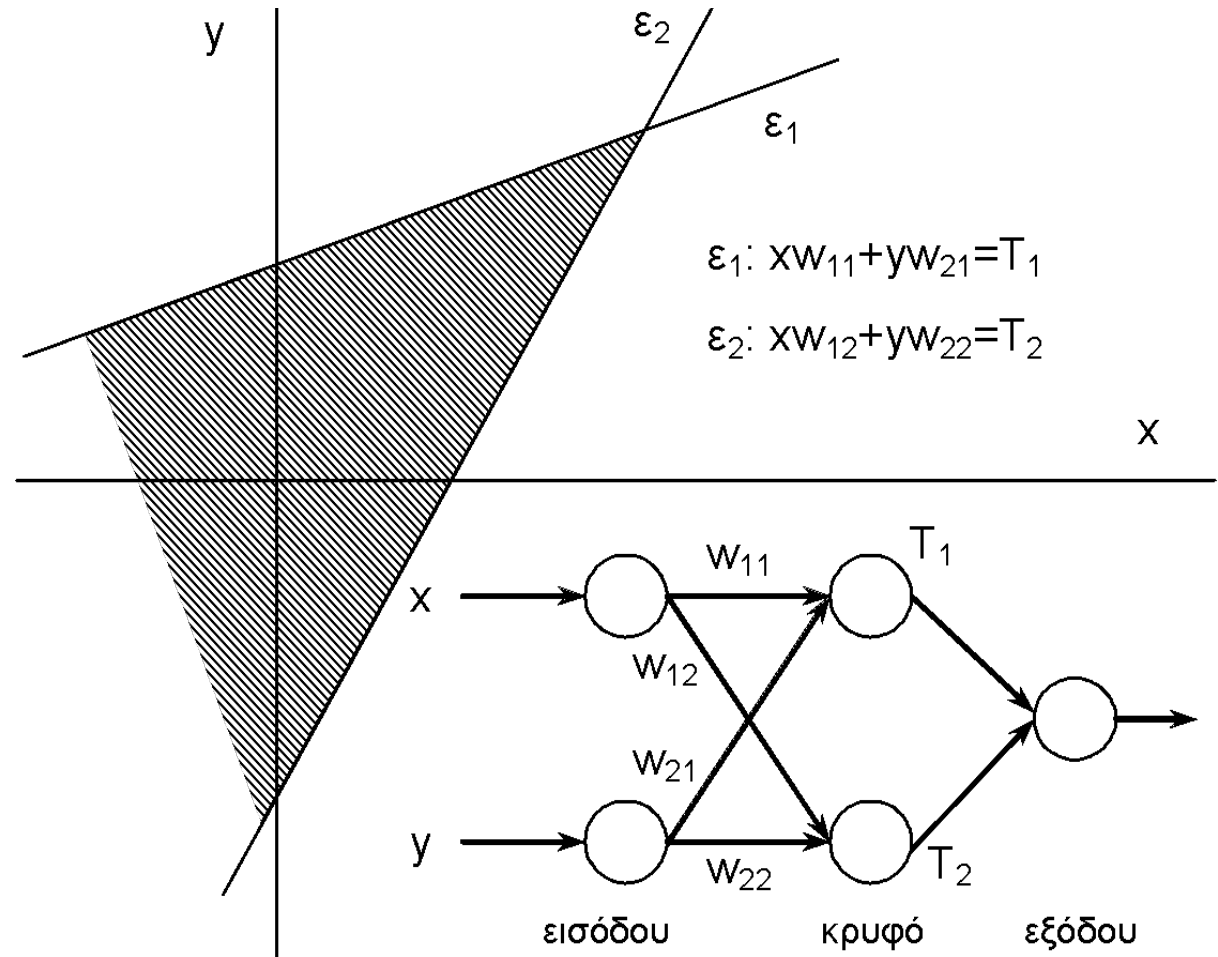
1. Υπολόγισε την έξοδο y
2. Εάν $y=t$ τότε δε γίνεται καμία μεταβολή στα βάρη
3. Εάν $y \neq t$ τότε μετάβαλε τα βάρη των ενεργών γραμμών εισόδου (αυτών που έχουν

σήμα $\neq 0$) κατά την ποσότητα $\Delta w = d \cdot (t - y) \cdot x$ έτσι ώστε το y να πλησιάσει το t .

❖ d : ρυθμός μάθησης (*learning rate*)

○ συνήθως παίρνει τιμή μεταξύ 0 και 1

❖ $t-y$: σφάλμα (*error*)



Παράδειγμα Μάθησης σε Perceptron

($\delta=0.2$, $T=0.5$)

Input		Initial weights		Output		Error	Final weights	
X ₁	X ₂	W ₁	W ₂	Desired (Z)	Network (Y)	(delta)	W ₁	W ₂
0	0	0.1	0.3	0	0	0	0.1	0.3
0	1	0.1	0.3	1	0	1	0.1	0.5
1	0	0.1	0.5	1	0	1	0.3	0.5
1	1	0.3	0.5	1	1	0	0.3	0.5
0	0	0.3	0.5	0	0	0	0.3	0.5
0	1	0.3	0.5	1	0	1	0.3	0.7
1	0	0.3	0.7	1	0	1	0.5	0.7
1	1	0.5	0.7	1	1	0	0.5	0.7
0	0	0.5	0.7	0	0	0	0.5	0.7
0	1	0.5	0.7	1	1	0	0.5	0.7
1	0	0.5	0.7	1	0	1	0.7	0.7
1	1	0.7	0.7	1	1	0	0.7	0.7

Κανόνας Δέλτα

- ❖ γενίκευση του αλγορίθμου εκπαίδευσης του perceptron
- ❑ ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος των διανυσμάτων εκπαίδευσης
- ❑ δεν μπορεί να εφαρμοστεί σε δίκτυα με κρυφά επίπεδα

❖ Μέσο τετραγωνικό σφάλμα E στο στοιχειώδες perceptron, για p διανύσματα εκπαίδευσης:

$$E = \frac{1}{p} \sum_{k=1}^p (t_k - f(\text{input}_k))^2$$

- ❑ input_k : σήμα εισόδου του νευρώνα, f : συνάρτηση ενεργοποίησης
- Συνολικό σήμα εισόδου για κάποιο διάνυσμα εκπαίδευσης p :

$$\text{input} = \sum_{i=1}^n w_i x_i$$

- ❑ t_k : επιθυμητή έξοδος

Κανόνας Δέλτα (2/3)

- ❖ Ο κανόνας Δέλτα ακολουθεί την αρνητική κλίση της επιφάνειας σφάλματος, με κατεύθυνση προς το ελάχιστό της:

$$\Delta w_i \propto -\frac{\partial E}{\partial w_i}$$

- ❖ Η παράγωγος του E ως προς τα w είναι:

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right)$$

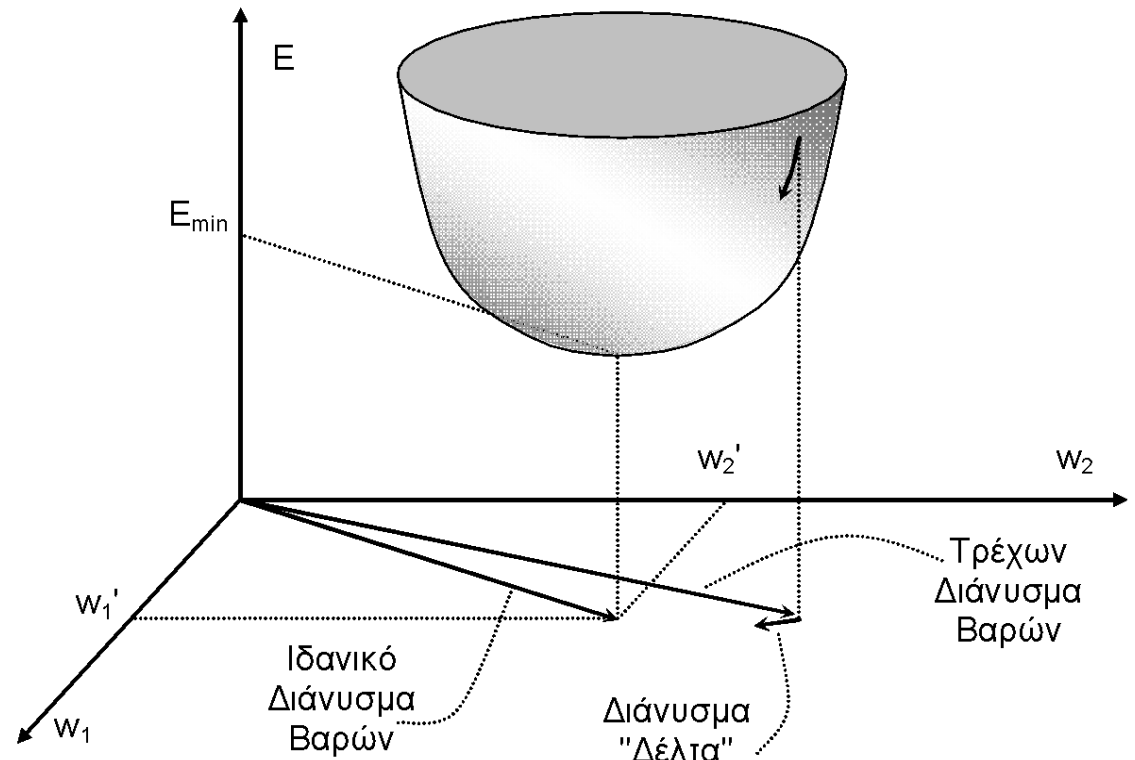
- ❖ Η μεταβολή στην τιμή του βάρους w_i , εξαιτίας της εκπαίδευσης με ένα μόνο από τα διανύσματα εκπαίδευσης, δίνεται από τη σχέση:

$$\Delta w_i = w_{i(new)} - w_{i(old)} = d(t - input)x_i$$

- ❖ Σε περίπτωση χρήσης συνεχούς συνάρτησης ενεργοποίησης (π.χ. σιγμοειδούς), προσθέτουμε έναν ακόμη όρο: την πρώτη παράγωγο f' της συνάρτησης f
 - Ο όρος αυτός εκφυλίζεται στη περίπτωση της βηματικής συνάρτησης

Κανόνας Δέλτα (3/3)

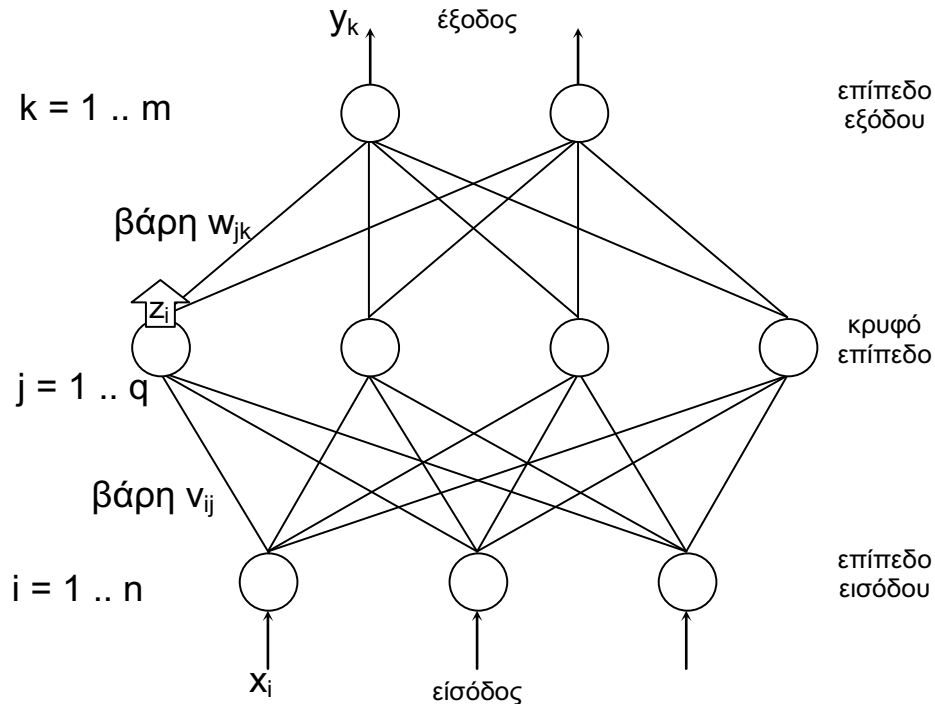
- ❖ γεωμετρική αναπαράσταση κανόνα Δέλτα για στοιχειώδες perceptron με 2 εισόδους
- ❖ κανόνας της επικλινοῦς καθόδου (*gradient descent rule*)
- ❖ δεν μπορεί να εφαρμοστεί σε δίκτυα με κρυφά επίπεδα
- ❖ **Αποτέλεσμα:** Η έρευνα στα ΤΝΔ περιορίστηκε σημαντικά για πολλά χρόνια.
- ❖ Η λύση δόθηκε με τη **μέθοδο της ανάστροφης μετάδοσης του λάθους**



Back Propagation (1/2)

❖ Βασίζεται στο γενικευμένο κανόνα Δέλτα - ΓΚΔ (*generalized Delta rule*)

□ **Βασική Ιδέα:** να καθοριστεί το ποσοστό του συνολικού σφάλματος που αντιστοιχεί σε κάθε νευρώνα, ακόμη και αυτών που ανήκουν σε κρυφά επίπεδα.



$$input_j = \sum_{i=1}^n v_{ij} x_i \quad z_j = f(input_j) = f\left(\sum_{i=1}^n v_{ij} x_i\right)$$

$$input_k = \sum_{j=1}^q w_{jk} z_j \quad y_k = f(input_k) = f\left(\sum_{j=1}^q w_{jk} z_j\right)$$

❖ Αποδεικνύεται ότι (σχέσεις ΓΚΔ):

□ για επίπεδο εξόδου: $\Delta w_{jk} = d \cdot \delta_k \cdot z_j$ με

$$\delta_k = (t_k - y_k) f'(input_k)$$

□ για κρυφό επίπεδο: $\Delta w_{ij} = d \cdot \delta_j \cdot x_i$ με

$$\delta_j = f'(input_j) \sum_{k=1}^m \delta_k w_{jk}$$

Back Propagation (2/2)

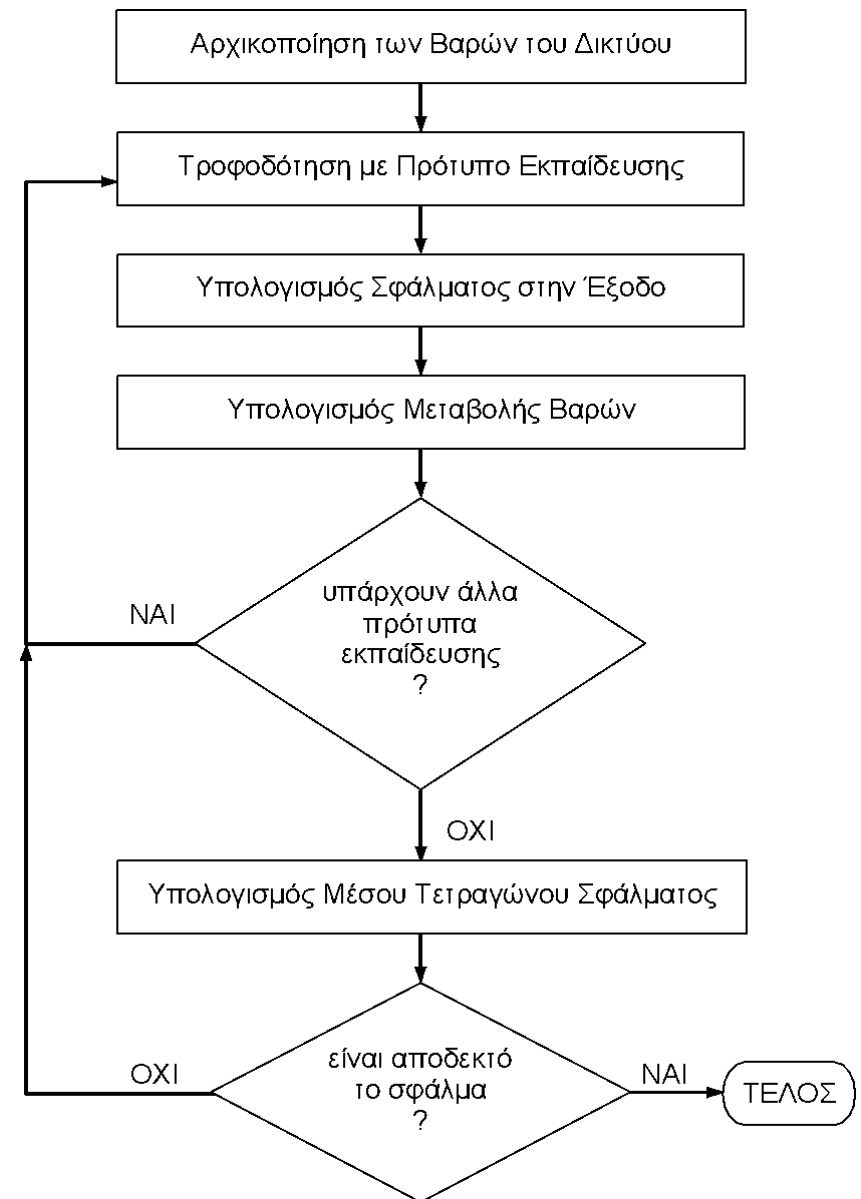
❖ Η αναπροσαρμογή των βαρών γίνεται από το επίπεδο εξόδου προς το εισόδου.

□ *ανάστροφο πέρασμα (backward pass) ή ανάστροφη μετάδοση (back propagation)*

❖ *Διαδικασία βελτιστοποίησης επικλινούς καθόδου (gradient descent optimization procedure) που ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα E μεταξύ της εξόδου του δικτύου και της επιθυμητής εξόδου, για τα p διανύσματα εκπαίδευσης.*

❖ *Υπολογισμός μέσου τετραγωνικού σφάλματος*

$$E = \frac{1}{p} \sum_p \sum_{k=1}^m (t_{k(p)} - y_{k(p)})^2$$



- ❖ Έστω ΤΝΔ με συνάρτηση ενεργοποίησης $g(z)$ που παίρνει πραγματικές τιμές
- ❖ Έξοδος νευρώνα i

$$a_i = g \bullet \left(\sum_{j=0}^n w_j a_j \right)$$

- ❖ Όπου a_i είναι η έξοδος του νευρώνα i , w_j το βάρος της σύνδεσης j και a_j η είσοδος j από τον νευρώνα του προηγούμενου επιπέδου.
- ❖ Η μεταβολή του βάρους w_j υπολογίζεται από τον τύπο:

$$w_j = w_{j_old} - d \bullet (a_i - a) a_j$$

- ❖ Όπου a είναι η επιθυμητή έξοδος του νευρώνα και d ο ρυθμός μάθησης.
- ❖ Το συνολικό σφάλμα υπολογίζεται:

$$Err = \frac{1}{2} \sum (a_i - a)^2$$

Μάθηση χωρίς Επίβλεψη

❖ Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.

- These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute. We want to explore the data to find some intrinsic structures in them.

Clustering is a technique for finding similarity groups in data, called clusters.

I.e., it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.

- Clustering is often called an unsupervised learning task as no class values denoting an a priori grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
- <https://www.youtube.com/watch?v=Ex0nhOPzLIs>

K-means algorithm

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be
 - $\{x_1, x_2, \dots, x_n\}$,
 - where $x_i = (x_{i1}, x_{i2}, \dots, x_{ir})$ is a vector in a real-valued space $X \subseteq \mathbb{R}^r$, and r is the number of attributes (dimensions) in the data.
- The k-means algorithm partitions the given data into k clusters.
- Each cluster has a cluster center, called centroid.
 - k is specified by the user
-
- Given k , the k-means algorithm works as follows:
 1. Randomly choose k data points (seeds) to be the initial centroids, cluster centers
 2. Assign each data point to the closest centroid
 3. Re-compute the centroids using the current cluster memberships.
 4. If a convergence criterion is not met, go to 2).



Algorithm k -means(k, D)

- 1 Choose k data points as the initial centroids (cluster centers)
- 2 **repeat**
- 3 **for** each data point $\mathbf{x} \in D$ do
- 4 compute the distance from \mathbf{x} to each centroid;
- 5 assign \mathbf{x} to the closest centroid // a centroid represents a cluster
- 6 **endfor**
- 7 re-compute the centroids using the current cluster memberships
- 8 **until** the stopping criterion is met

Stopping/convergence criterion

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the sum of squared error (SSE),

$$SSE = \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} \text{dist}(\mathbf{x}, \mathbf{m}_j)^2$$

C_j is the j^{th} cluster, \mathbf{m}_j is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $\text{dist}(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point \mathbf{x} and centroid \mathbf{m}_j .

The k -means algorithm can be used for any application data set where the **mean** can be defined and computed. In the **Euclidean space**, the mean of a cluster is computed with:

$$\mathbf{m}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \quad (2)$$

where $|C_j|$ is the number of data points in cluster C_j . The distance from one data point \mathbf{x}_i to a mean (centroid) \mathbf{m}_j is computed with

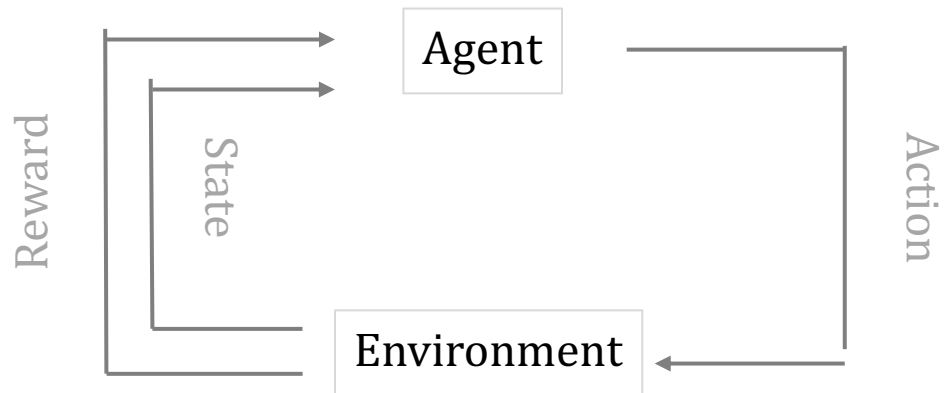
$$\begin{aligned} \text{dist}(\mathbf{x}_i, \mathbf{m}_j) &= \|\mathbf{x}_i - \mathbf{m}_j\| \\ &= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \dots + (x_{ir} - m_{jr})^2} \end{aligned} \quad (3)$$

Εφαρμογές Νευρωνικών Δικτύων

- ❖ Δημοφιλή σε προβλήματα που περιέχουν μη-προβλέψιμες λειτουργίες και τα οποία δεν είναι πλήρως κατανοητά.
- ❖ Κατηγοριοποίηση
 - ❑ Ιατρική, Άμυνα, Γεωργία, Οικονομία/επιχειρήσεις
- ❖ Αναγνώριση Προτύπων
 - ❑ Τράπεζες, Πληροφορική και Τηλεπικοινωνίες
- ❖ Αποτίμηση
 - ❑ Άμυνα, Ασφάλεια, Μηχανολογία
- ❖ Πρόβλεψη
 - ❑ Οικονομία/επιχειρήσεις, Γεωργία, Μετεωρολογία

Επαυξητική Μάθηση

- Learning to interact with an environment
 - Robots, games, process control
 - With limited human training
 - Where the ‘right thing’ isn’t obvious
- Supervised Learning:
 - Goal: $f(x)=y$
 - Data: [$\langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle$]
- Reinforcement Learning:
 - Goal:
 - Maximize $\sum_{i=1}^{\infty} \text{Reward}(\text{State}_i, \text{Action}_i)$
- Data: $\text{Reward}_i, \text{State}_{i+1} = \text{Interact}(\text{State}_i, \text{Action}_i)$



Markov Decision Process

- Discrete-time stochastic control process
- Each time step, s :
 - Agent chooses action a from set A_s
 - Moves to new state with probability:
 - $P_a(s, s')$
 - Receives reward: $R_a(s, s')$
- Every outcome depends on s and a
 - Nothing depends on previous states/actions

Policy

- $\pi(s)$: the action to take in state s
 - goal maximise: $\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$
 - $a_t = \pi(s_t)$
 - $0 \leq \gamma < 1$ – Tradeoff immediate vs future

- $V^{\pi}(s) = \sum_{s'} P_{\pi(s)}(s, s') * (R_{\pi(s)}(s, s') + \gamma V^{\pi}(s'))$

Reward for making that move

Value of being in that state

Probability of moving to each state

Q learning

- Learn a policy $\pi(s)$ that optimizes $V^\pi(s)$ for all states, using:
 - No prior knowledge of state transition probabilities: $P_a(s, s')$
 - No prior knowledge of the reward function: $R_a(s, s')$
- Approach (assume a learning rate γ e.g. 0.8):
 - Initialize estimate of discounted reward for every state/action pair: $\hat{Q}(s, a) = 0$
 - Repeat (for a while):
 - Take a random action a from A_s
 - Receive s' and $R_a(s, s')$ from environment
 - Update $\hat{Q}(s, a) = R_a(s, s') + \gamma \max_{a'} \hat{Q}(s', a')$
 - Random restart if in terminal state

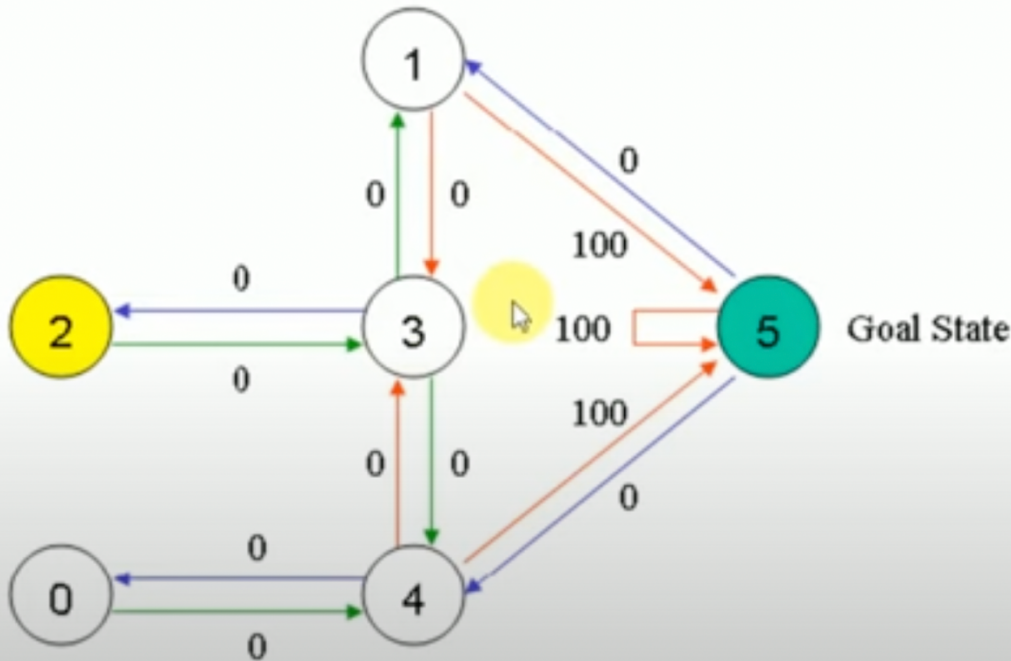
$$\alpha_v = \frac{1}{1 + \text{visits}(s, a)}$$

$$\text{Exploration Policy: } P(a_i, s) = \frac{k \hat{Q}(s, a_i)}{\sum_j k \hat{Q}(s, a_j)}$$

Παράδειγμα

<https://www.youtube.com/watch?v=J3qX50yyiU0>

We can put the state diagram and the instant reward values into the following reward table, "matrix R". The -1's in the table represent null values (i.e.; where there isn't a link between nodes). For example, State 0 cannot go to State 1.



State Action

State	0	1	2	3	4	5
0	-1	-1	-1	-1	0	-1
1	-1	-1	-1	0	-1	100
2	-1	-1	-1	0	-1	-1
3	-1	0	0	-1	0	-1
4	0	-1	-1	0	-1	100
5	-1	0	-1	-1	0	100

$R =$

Look at the second row (state 1) of matrix R.

There are two possible actions for the current state 1: go to state 3, or go to state 5.

By random selection, we select to go to 5 as our action.

	Action					
State	0	1	2	3	4	5
0	-1	-1	-1	-1	0	-1
1	-1	-1	-1	0	-1	100
2	-1	-1	-1	0	-1	-1
3	-1	0	0	-1	0	-1
4	0	-1	-1	0	-1	100
5	-1	0	-1	-1	0	100

Now let's imagine what would happen if our agent were in state 5.

Look at the sixth row of the reward matrix R (i.e. state 5).

It has 3 possible actions: go to state 1, 4 or 5.

$$Q(\text{state}, \text{action}) = R(\text{state}, \text{action}) + \text{Gamma} * \text{Max}[Q(\text{next state}, \text{all actions})]$$

$$Q(1, 5) = R(1, 5) + 0.8 * \text{Max}[Q(5, 1), Q(5, 4), Q(5, 5)] = 100 + 0.8 * 0 = 100$$

		Action					
State		0	1	2	3	4	5
$R =$	0	-1	-1	-1	-1	0	-1
	1	-1	-1	-1	0	-1	100
	2	-1	-1	-1	0	-1	-1
	3	-1	0	0	-1	0	-1
	4	0	-1	-1	0	-1	100
	5	1	0	1	1	0	100

$Q =$	0	0	0	0	0	0	0
	1	0	0	0	0	0	100
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

For the next episode, we randomly choose the initial state – say 3 (can go to 1, 2 & 4)

$$R = \begin{array}{c|cccccc} & \text{Action} & & & & & \\ \text{State} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & -1 & -1 & -1 & -1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 0 & -1 & 100 \\ 2 & -1 & -1 & -1 & 0 & -1 & -1 \\ 3 & -1 & 0 & 0 & -1 & 0 & -1 \\ 4 & 0 & -1 & -1 & 0 & -1 & 100 \\ 5 & -1 & 0 & -1 & -1 & 0 & 100 \end{array}$$

Now we imagine that we are in state 1 (next state).

Now we imagine that we are in state 1 (next state).

Look at the second row of reward matrix R (i.e. state 1).

It has 2 possible actions: go to state 3 or state 5.

Then, we compute the Q value:

$$Q(\text{state}, \text{action}) = R(\text{state}, \text{action}) + \text{Gamma} * \text{Max}[Q(\text{next state}, \text{all actions})]$$

$$Q(3, 1) = R(3, 1) + 0.8 * \text{Max}[Q(1, 3), Q(1, 5)] = 0 + 0.8 * \text{Max}(0, 100) = 80$$

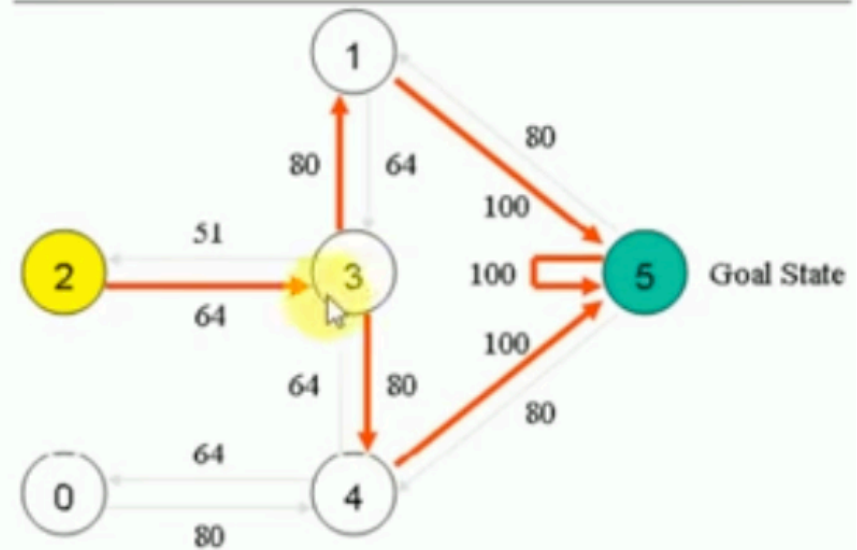
		Action					
State		0	1	2	3	4	5
$R =$	0	-1	-1	-1	-1	0	-1
	1	-1	-1	-1	0	-1	100
	2	-1	-1	-1	0	-1	-1
	3	-1	0	0	-1	0	-1
	4	0	-1	-1	0	-1	100
	5	-1	0	-1	-1	0	100

$Q =$	0	0	0	0	0	0	0
	1	0	0	0	0	0	100
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

If our agent learns more through further episodes, it will finally reach convergence values in matrix Q like:

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{matrix}$$

Tracing the best sequences of states is as simple as following the links with the highest values at each state.



Αναπαράσταση Γνώσης

Σύνολο συντακτικών και σημασιολογικών παραδοχών, οι οποίες καθιστούν δυνατή την περιγραφή ενός κόσμου.

- ❖ Η φυσική γλώσσα είναι ακατάλληλη για αναπαράσταση γνώσης
- ❑ Πολυσημαντικότητα (ambiguity)
- ❑ Ερμηνεία με βάση τα συμφραζόμενα (context)
- ❖ Χρειάζεται μονοσήμαντος και τυποποιημένος συμβολισμός
- ❑ Επακριβής αναπαράσταση γνώσης
- ❑ Συνδυασμός με μηχανισμό εξαγωγής συμπερασμάτων (**inference mechanism**)
- ❖ Μία μέθοδος αναπαράστασης γνώσης έχει:
 - ❑ *Συντακτικό (syntax)*: Σύμβολα και κανόνες συνδυασμού τους
 - ❑ *Σημασιολογία (semantics)*: Καθορισμός των εννοιών που αποδίδονται στα σύμβολα και στους συνδυασμούς που επιτρέπει το συντακτικό.

Δεδομένα, Πληροφορία, Γνώση και Σοφία



Δεδομένα

- ❖ Μη-οργανωμένα και μη-επεξεργασμένα γεγονότα σχετικά με αντικείμενα ή συμβάντα του πραγματικού κόσμου
- ❖ Μετρήσιμες ή υπολογίσιμες τιμές των ιδιοτήτων των αντικειμένων
 - ❑ Π.χ. η τιμή πώλησης ενός προϊόντος σε μία εμπορική συναλλαγή
 - ❑ Π.χ. η ημερομηνία γέννησης ενός ατόμου
- ❖ Είναι στατικά
 - ❑ Από τη στιγμή που θα καταγραφούν δεν αλλάζουν
- ❖ Χωρίς ένα πλαίσιο αναφοράς, δεν είναι ιδιαίτερα χρήσιμα
 - ❑ Δεν αποτελούν πληροφορία

Πληροφορία

- ❖ Χρειάζεται ένα εννοιολογικό πεδίο που να επιτρέπει την ερμηνεία της
- ❖ Αποτελείται από δεδομένα που έχουν φιλτραριστεί και μορφοποιηθεί κατάλληλα
 - ❑ Π.χ., ετήσιο άθροισμα πωλήσεων κάποιου υποκαταστήματος επιχείρησης
 - ❑ Επεξεργασμένη πληροφορία, λόγω της άθροισης
 - ❑ Φιλτραρισμένη πληροφορία, λόγω του συγκεκριμένου υποκαταστήματος

❖ Η πληροφορία, σε σύγκριση με τα δεδομένα, έχει νόημα, σκοπό και συνάφεια, ώστε να μπορεί να ερμηνεύεται και να διευκολύνει στη λήψη αποφάσεων.

Γνώση

- ❖ Πληροφορία που έχει υποστεί μία σειρά ειδικών ελέγχων για την πιστοποίησή της
 - ❑ Π.χ. επιστημονική γνώση
 - ❑ Π.χ. γνώση που προέρχεται από μακρόχρονη επιβεβαίωση των καθημερινών εμπειριών
 - ❑ Π.χ. η επαναλαμβανόμενη παρατήρηση πως οι πωλήσεις από ένα συγκεκριμένο υποκατάστημα αυξάνονται 20% κατά τους καλοκαιρινούς μήνες
- ❖ Αναδεικνύει τη σημαντικότητα της πληροφορίας συσχετίζοντάς τη με χρήσιμα συμπεράσματα ή αναγκαίες ενέργειες.

Γνώση θεωρείται η κατανόηση που αποκτάται μέσω εμπειρίας ή μελέτης και συμπεριλαμβάνει όλες τις πληροφορίες, τις εμπειρίες, τις ικανότητες, τις δεξιότητες και την κοινή λογική, που κατέχει ένας άνθρωπος.

- ❖ Έχει συνήθως αφηρημένη ή γενικευμένη μορφή.
 - ❑ Π.χ. όλοι οι ρόμβοι είναι τετράπλευρα.
- ❖ ποιότητα γνώσης = ποιότητα πληροφορίας + αξιοπιστία κανόνων πιστοποίησης

Σοφία (*wisdom*): Ικανότητα να χρησιμοποιεί κάποιος τη γνώση όσο το δυνατόν αποδοτικότερα (αναθεώρηση, μάθηση, διορατικότητα, πρόβλεψη).

Είδη Γνώσης

- ❖ **Αντικείμενα (objects):** Αναπαράσταση των αντικειμένων ενός κόσμου και της σχέσης τους
 - ❑ Σημασιολογική γνώση (semantic knowledge), ιεραρχικά δομημένη.
- ❖ **Γεγονότα (events):** Αναπαράσταση των ενεργειών και της χρονικής ακολουθίας με την οποία συμβαίνουν, καθώς και τις σχέσεις αίτιου-αποτελέσματος.
 - ❑ Επεισοδιακή γνώση (episodical knowledge).
 - ❑ Προσωπικές εμπειρίες ενός ατόμου, οργανωμένες χρονικά και χωρικά σε επεισόδια και όχι σε έννοιες ή σχέσεις.
- ❖ **Εκτέλεση (performance):** Αναπαράσταση των δεξιοτήτων για το πώς κάποιος κάνει πράγματα (εκτελεί εργασία ή διεκπεραιώνει διαδικασία)
 - ❑ Διαδικαστική γνώση (procedural knowledge).
- ❖ **Μετα-γνώση (meta-knowledge):** Αναπαράσταση της γνώσης για το τι γνωρίζει κάποιος και πότε πρέπει να το εφαρμόσει.
 - ❑ Συνώνυμη της σοφίας.

Κριτήρια Αξιολόγησης Μεθόδων Αναπαράστασης Γνώσης

- ❖ Επάρκεια αναπαράστασης (representational adequacy).
 - ❑ Η ικανότητα να αναπαριστά όλα τα είδη της γνώσης.

- ❖ Επάρκεια συνεπαγωγής (inferential adequacy).
 - ❑ Η ικανότητα να συνεργάζεται με μηχανισμούς που επεξεργάζονται υπάρχουσες δομές γνώσης.

- ❖ Αποδοτικότητα συνεπαγωγής (inferential efficiency).
 - ❑ Η ικανότητα να μπορεί να εισάγει επιπλέον πληροφορίες στις δομές γνώσης, έτσι ώστε να κατευθύνει τους μηχανισμούς εξαγωγής συμπερασμάτων γρηγορότερα προς τη λύση.

- ❖ Αποδοτικότητα απόκτησης (acquisitional efficiency).
 - ❑ Η ικανότητα να επιτρέπει την απόκτηση νέας γνώσης εύκολα και γρήγορα.

Διαδικαστική Αναπαράσταση

- ❖ Αναπαράσταση γνώσης μέσω εξειδικευμένων διαδικασιών,
 - ❑ Κωδικοποίηση σε γλώσσα προγραμματισμού
 - ❑ Επίλυση συγκεκριμένων υποπροβλημάτων
 - ❑ Ενεργοποιούνται είτε από άλλες διαδικασίες είτε από δεδομένα
 - ❑ *Δαίμονες* (demons): Ενεργοποιούνται μόλις ικανοποιηθεί μια συνθήκη
- ❖ Ανάμιξη της γνώσης ελέγχου για την επίλυση του προβλήματος με την ίδια την γνώση για το πρόβλημα, το οποίο προκαλεί.
 - ❑ **Πλεονέκτημα**: Γρηγορότερη επεξεργασία της κωδικοποιημένης γνώσης
 - ❑ **Μειονέκτημα**: Αυξημένη δυσκολία αναθεώρησης ή προσθήκης γνώσης
 - ❑ **Μειονέκτημα**: Μικρότερη διαφάνεια του προγράμματος
- ❖ Συνήθως απαιτείται συνδυασμός διαδικαστικών και δηλωτικών μεθόδων.

Συλλογιστική (reasoning)

❖ Αναπόσπαστο συστατικό της νοημοσύνης

Συλλογιστική: Μέθοδος με την οποία τμήματα υπάρχουσας γνώσης συνδυάζονται μεταξύ τους ώστε να παράγουν νέα γνώση ή να εξάγουν συμπεράσματα.

- ❖ Κάθε μέθοδος αναπαράστασης της γνώσης έχει τις δικές της συλλογιστικές.
- ❖ Πιο γνωστές συλλογιστικές (για συστήματα κανόνων):
 - ❑ Συνεπαγωγή (*deduction*)
 - ❑ Επαγωγή (*induction*)
 - ❑ Απαγωγή (*abduction*)
- ❖ Εξελιγμένες συλλογιστικές που χρησιμοποιούνται στα συστήματα γνώσης:
 - ❑ Συλλογιστική των Μοντέλων (*model-based reasoning*)
 - ❑ Ποιοτική Συλλογιστική (*qualitative reasoning*)
 - ❑ Συλλογιστική των Περιπτώσεων (*case-based reasoning*)
 - Συλλογιστική με αναλογίες (*analogical reasoning*)

Εξαγωγή Συμπερασμάτων (inference)

- ❖ Εξαρτάται από:
 - ❑ τη συλλογιστική
 - ❑ τη στρατηγική αναζήτησης στη γνώση ενός προβλήματος
- ❖ *Στρατηγική αναζήτησης*: Ο τρόπος με τον οποίο έχει δομηθεί και κωδικοποιηθεί η γνώση προκειμένου να δώσει λύση σε ένα πρόβλημα.
- ❖ Η **στρατηγική αναζήτησης** υλοποιείται με διάφορους τρόπους:
 - ❑ **Οδηγούμενη από τους στόχους** (goal driven ή top-down): Ξεκινάμε από πιθανά συμπεράσματα και φτάνουμε στις αιτίες που τα στηρίζουν.
 - ❑ **Οδηγούμενη από τα δεδομένα** (data driven ή bottom-up): Ξεκινάμε από τα δεδομένα του προβλήματος και φτάνουμε σε συμπεράσματα.
- ❖ Οι **συλλογιστικές** υλοποιούνται από έναν ή περισσότερους εναλλακτικούς μηχανισμούς εξαγωγής συμπερασμάτων (*inference mechanisms*)

Μηχανισμός Εξαγωγής Συμπερασμάτων: Αλγόριθμος που συνδυάζει τις δομές που αναπαριστούν τα διάφορα τμήματα της γνώσης που βρίσκονται αποθηκευμένα στη μνήμη του υπολογιστή, σύμφωνα με το αφαιρετικό μοντέλο της συλλογιστικής, και παράγει νέες δομές που επίσης αποθηκεύει στον υπολογιστή.

Είδη Συλλογιστικής

Συνεπαγωγική συλλογιστική (*deductive reasoning*): Εξάγει συμπεράσματα βασισμένα στους κλασικούς μηχανισμούς εξαγωγής συμπερασμάτων της λογικής.

Δεδομένου του κανόνα: Όλα τα σκυλιά του Κώστα είναι καφέ
και του γεγονότος: Αυτά τα σκυλιά είναι του Κώστα
Συμπέρασμα που εξάγεται: Αυτά τα σκυλιά είναι καφέ

Επαγωγική συλλογιστική (*inductive reasoning*): αφορά την εξαγωγή γενικών συμπερασμάτων από ένα σύνολο παραδειγμάτων.

Δεδομένων των γεγονότων: Το σκυλί Α είναι του Κώστα και είναι καφέ .

Το σκυλί Β είναι του Κώστα και είναι καφέ .

...

Κανόνας που εξάγεται: Όλα τα σκυλιά του Κώστα είναι καφέ .

Απαγωγική συλλογιστική (*abductive reasoning*): Εξαγωγή συμπερασμάτων κατά την οποία, με δεδομένα μία βάση γνώσης και μερικές παρατηρήσεις (observations) επιχειρείται η εύρεση υποθέσεων οι οποίες μαζί με τη βάση γνώσης εξηγούν τις παρατηρήσεις.

Δεδομένου του κανόνα: Όλα τα σκυλιά του Κώστα είναι καφέ
και του αποτελέσματος: Τα σκυλιά είναι καφέ
Υπόθεση που γίνεται: Αυτά τα σκυλιά είναι του Κώστα

Σύγκριση Συλλογιστικών Επαγωγής vs. Απαγωγής

- ❖ Και οι 2 εμπεριέχουν τη δημιουργία και τον έλεγχο υποθέσεων.
- ❖ Στην επαγωγή, η υπόθεση αφορά ένα γενικό κανόνα που εξηγεί τα γεγονότα
 - ❑ Η ορθότητα του κανόνα-υπόθεση απαιτεί μεγάλο αριθμό παρόμοιων καταστάσεων
- ❖ Στην απαγωγή η υπόθεση αφορά ένα συγκεκριμένο γεγονός
 - ❑ Για την εξαγωγή συμπεράσματος αρκεί μία μόνο κατάσταση

Συνεπαγωγή vs. Απαγωγή

- ❖ Στη συνεπαγωγή, το συμπέρασμα που εξάγεται είναι λογικό επακόλουθο του γενικού κανόνα και του γεγονότος που καταγράφεται ως αληθές
 - ❑ Όταν ο γενικός κανόνας δεν είναι απολύτως βέβαιος, τότε δεν μπορεί να εφαρμοστεί
 - ❑ Π.χ. Σχεδόν όλα τα σκυλιά του Κώστα είναι καφέ
- ❖ Στην απαγωγή, η υπόθεση είναι **μία** από τις πιθανές και δεν είναι απόλυτα αληθής
 - ❑ Όταν ο γενικός κανόνας δεν είναι απολύτως βέβαιος, εξακολουθεί να είναι εφαρμόσιμη
 - ❑ Το γεγονός-υπόθεση εξακολουθεί να αποτελεί μια πιθανή εξήγηση της παρατήρησης

Λογική

- ❖ Αποσαφήνιση και την τυποποίηση της διαδικασίας της ανθρώπινης σκέψης.
- ❖ Η μαθηματική λογική (*mathematical logic*) είναι η συστηματική μελέτη των *έγκυρων ισχυρισμών* (*valid arguments*).

□ Ένας ισχυρισμός (*argument*) αποτελείται από συγκεκριμένες δηλώσεις (ή προτάσεις), τις υποθέσεις (*premises*), και τα συμπεράσματα (*conclusions*).

Όλοι οι άνθρωποι είναι θνητοί,

(Δήλωση)

Ο Σωκράτης είναι άνθρωπος,

(Δήλωση)

επομένως, ο Σωκράτης είναι θνητός

(Συμπέρασμα)

- ❖ Συμβολική λογική (*symbolic logic*).

□ Οι ισχυρισμοί μελετώνται ανεξάρτητα από το θέμα το οποίο πραγματεύονται.

P: $\forall X$ άνθρωπος (X) \rightarrow θνητός (X) .

Q: άνθρωπος (Σωκράτης)

R: θνητός (Σωκράτης) .

$P \wedge Q \models R$

Σύνταξη και Σημασιολογία

- ❖ Απαιτείται ο ορισμός της *σύνταξης* (*syntax*) και της *σημασιολογίας* (*semantics*).
- ❖ Η *σύνταξη* καθορίζει τις επιτρεπτές ακολουθίες συμβόλων.
- ❖ Η *σημασιολογία* καθορίζει τις μεταξύ τους σχέσεις.
- ❖ Η *ερμηνεία*
 - ❑ αντιστοιχεί τα σύμβολα της γλώσσας στις οντότητες του κόσμου που αναπαρίσταται
 - ❑ επιτρέπει την απόδοση λογικών τιμών στις προτάσεις της γλώσσας.

Προτασιακή Λογική

- ❖ Στην προτασιακή λογική κάθε γεγονός του πραγματικού κόσμου
 - αναπαριστάται με μια λογική πρόταση,
 - χαρακτηρίζεται είτε ως αληθής (*T-true*) ή ως ψευδής (*F-false*)
- ❖ Οι λογικές προτάσεις (*άτομα - atoms*) αναπαριστώνται συνήθως από λατινικούς χαρακτήρες.
- ❖ Συνδυάζονται με τη χρήση λογικών συμβόλων ή συνδετικών (*connectives*)

Σύμβολο	Ονομασία / Επεξήγηση
\wedge	σύζευξη (λογικό "ΚΑΙ")
\vee	διάζευξη (λογικό "Η")
\neg	άρνηση
\rightarrow	συνεπαγωγή ("ΕΑΝ ΤΟΤΕ")
\leftrightarrow	διπλή συνεπαγωγή ή ισοδυναμία ("ΑΝ ΚΑΙ ΜΟΝΟ ΑΝ").

- ❖ Τρία σημεία στίξης
 - Δύο Παρενθέσεις "(" , ")" και το κόμμα ","

❖ Ορθά δομημένοι τύποι (well formed formulae).

Παράδειγμα Χρήσης Συνδετικών

- ❖ Αναπαράσταση της ακόλουθης γνώσης με προτασιακή λογική:
 - 1^η πρόταση: "επιδιώκω την ειρήνη"
 - 2^η πρόταση: "αποφεύγω πόλεμο"
 - 3^η πρόταση: "εάν επιδιώκω την ειρήνη, τότε αποφεύγω πόλεμο"
- ❖ Σε κάθε πρόταση αντιστοιχεί ένας λατινικός χαρακτήρας.
 - P: "επιδιώκω την ειρήνη"
 - Q: "αποφεύγω τον πόλεμο"
- ❖ Η 3^η πρόταση αναπαριστάται με την χρήση του συνδετικού της συνεπαγωγής:
 $P \rightarrow Q$ "εάν επιδιώκω την ειρήνη, τότε αποφεύγω τον πόλεμο"

Σημασιολογία της Προτασιακής Λογικής

❖ Αντιστοιχεί μία τιμή αληθείας (αληθές T ή ψευδές F) σ' έναν τύπο, βασισμένη σε μια ερμηνεία της γλώσσας.

❖ Μια ερμηνεία (*interpretation*)

□ αντιστοιχεί τιμές αληθείας στα άτομα, και

□ επεκτείνεται σε σύνθετους τύπους με χρήση ενός πίνακα αληθείας (truth table).

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

❖ Έστω η ερμηνεία $I = \{I(P) = \mathbf{T}, I(Q) = \mathbf{T}\}$.

□ Σύμφωνα με αυτή την ερμηνεία και τον πίνακα αλήθειας ο τύπος $P \rightarrow Q$ είναι αληθής.

□ Ο τύπος $P \rightarrow Q$ ικανοποιείται από την ερμηνεία I . Τότε λέμε ότι η ερμηνεία αυτή αποτελεί *Μοντέλο (model)* του τύπου.

Ενδιαφέρουσες Περιπτώσεις Τύπων

❖ *Ταυτολογία (tautology)*: αληθής κάτω από οποιαδήποτε ερμηνεία.

□ $P \vee \neg P.$

□ Εάν ο τύπος F είναι ταυτολογία τότε γράφεται $\models F.$

❖ *Αντίφαση (contradiction)*: ψευδής κάτω από οποιαδήποτε ερμηνεία.

□ $P \wedge \neg P.$

❖ Ένας τύπος P συνεπάγεται λογικά (*implication*) από τον τύπο Q εάν κάθε μοντέλο του Q είναι επίσης και μοντέλο του $P.$

□ Η περίπτωση συμβολίζεται ως $Q \models P.$

❖ Δύο τύποι P και Q ονομάζονται *ισοδύναμοι (equivalent)* εάν οι πίνακες αλήθειας τους είναι οι ίδιοι κάτω από οποιαδήποτε ερμηνεία.

□ Η λογική ισοδυναμία ορίζεται με το σύμβολο \Leftrightarrow , πχ. $P \Leftrightarrow Q.$

Ορισμοί σε σύνολα τύπων

❖ Ένα σύνολο τύπων S ονομάζεται

□ *ταυτολογία*: κάθε ερμηνεία του συνόλου S ικανοποιεί κάθε τύπο του S .

□ *ικανοποιήσιμο (satisfiable)*: υπάρχει μια τουλάχιστον ερμηνεία που να ικανοποιεί όλους τους τύπους του S ,

□ *μη-ικανοποιήσιμο (unsatisfiable) ή αντίφαση*: δεν υπάρχει δυνατή ερμηνεία που να ικανοποιεί όλους τους τύπους του S .

❖ Μια πρόταση P *λογικά συνεπάγεται (implication ή entailment)* από ένα σύνολο S όταν κάθε ερμηνεία η οποία ικανοποιεί το S ικανοποιεί επίσης και το P και συμβολίζεται με $S \models P$.

❖ Δυο σύνολα προτάσεων S και F ονομάζονται *λογικά ισοδύναμα* εάν $S \models F$ και $F \models S$.

❖ Διαφορά της *λογικής ισοδυναμίας* και του *συνδετικού της ισοδυναμίας*

□ Η λογική ισοδυναμία (\Leftrightarrow) αφορά τη σημασιολογία των υπό εξέταση προτάσεων.

□ Το συνδετικό της ισοδυναμίας (\leftrightarrow) αποτελεί μέρος της σύνταξης της γλώσσας.

□ Το ίδιο ισχύει για τη λογική συνεπαγωγή (\vDash) και το συνδετικό της συνεπαγωγής (\rightarrow).

Παράδειγμα

❖ Παράδειγμα:

□ P: "επιδιώκω την ειρήνη"

□ Q: "αποφεύγω τον πόλεμο"

□ $P \rightarrow Q$ "εάν επιδιώκω την ειρήνη, τότε αποφεύγω τον πόλεμο"

❖ Έστω η ερμηνεία $I = \{I(P) = t, I(Q) = t\}$. Τότε ο τελευταίος τύπος είναι αληθής.

❖ Αντίθετα η δήλωση $P \wedge Q \not\models P \rightarrow Q$, δηλώνει ότι κάθε ερμηνεία που ικανοποιεί τον τύπο $P \wedge Q$ ικανοποιεί επίσης και τον τύπο $P \rightarrow Q$.

Λογικές Ισοδυναμίες

- ❖ Υπάρχει ένα σύνολο ισοδυναμιών που χρησιμοποιούνται για την μετατροπή μιας πρότασης σε κάποια ισοδύναμή της
- ❖ Οι ισοδυναμίες είναι αληθείς κάτω από οποιαδήποτε ερμηνεία.

	Ισοδυναμία	Ονομασία
(1)	$P \Leftrightarrow \neg\neg P$	νόμος της διπλής άρνησης
(2)	$(\neg P \vee \neg Q) \Leftrightarrow \neg(P \wedge Q)$	νόμος De Morgan
(3)	$(\neg P \wedge \neg Q) \Leftrightarrow \neg(P \vee Q)$	νόμος De Morgan
(4)	$(P \vee Q) \wedge R \Leftrightarrow (P \wedge R) \vee (Q \wedge R)$	επιμερισμός ως προς την σύζευξη
(5)	$(P \wedge Q) \vee R \Leftrightarrow (P \vee R) \wedge (Q \vee R)$	επιμερισμός ως προς την διάζευξη
(6)	$(P \rightarrow Q) \Leftrightarrow \neg P \vee Q$	Οποιοσδήποτε τύπος της προτασιακής λογικής μπορεί να μετατραπεί σε ένα ισοδύναμο χωρίς την χρήση των συνδετικών της συνεπαγωγής και της ισοδυναμίας
(7)	$(P \leftrightarrow Q) \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$	

Κανονικές Μορφές

- ❖ *Κανονικές μορφές (canonical forms)*: Μορφές των τύπων της λογικής
 - ❑ δεν εμφανίζονται καθόλου κάποια συνδετικά
 - ❑ ακολουθούν μια συγκεκριμένη δομή.
 - ❑ Π.χ. στη *Διαζευκτική* και *Συζευκτική* μορφή της λογικής, χρησιμοποιούνται μόνο τα συνδετικά της σύζευξης, διάζευξης και άρνησης.
- ❖ Κάθε τύπος μπορεί να μετατραπεί σε μια κανονική μορφή, χρησιμοποιώντας:
 - ❑ τις ισοδυναμίες για την απαλοιφή των συνδετικών της ισοδυναμίας και συνεπαγωγής
 - ❑ την κατάλληλη ομαδοποίηση των ατόμων μέσω των ισοδυναμιών του επιμερισμού
- ❖ Οι κανονικές μορφές της λογικής, είναι χρήσιμες για
 - ❑ την εύρεση της λογικής τιμής μιας πολύπλοκης έκφρασης.
 - ❑ την εξαγωγή νέας γνώσης.

Διαζευκτική Και Συζευκτική Κανονική Μορφή της Λογικής

❖ Στην *διαζευκτική κανονική μορφή της λογικής (disjunctive normal form)*, οι προτάσεις αποτελούνται από διαζεύξεις τύπων που μπορεί να είναι μόνο:

- ❑ λεκτικά (*literals*) και
- ❑ συζεύξεις λεκτικών

$$(Q \wedge R \wedge \neg S) \vee (V \wedge W) \vee (R \wedge S) \vee \dots \vee (X \wedge Z)$$

❖ Στην *συζευκτική μορφή της λογικής (conjunctive normal form)* οι προτάσεις αποτελούνται από συζεύξεις διαζεύξεων, δηλαδή έχουν την μορφή:

$$(Q \vee R \vee \neg S) \wedge (V \vee W) \wedge (R \vee S) \wedge \dots \wedge (X \vee Z)$$

Παράδειγμα Κανονικής Μορφής

❖ Έστω η ακόλουθη γνώση εκφρασμένη στη γενική μορφή της προτασιακής λογικής:
" επιδιώκω την ειρήνη" ΚΑΙ "εάν επιδιώκω την ειρήνη, τότε αποφεύγω τον πόλεμο"

❖ Σε συμβολική μορφή:

$$P \wedge (P \rightarrow Q)$$

❖ Σε κανονική διαζευκτική μορφή:

$$(P \wedge \neg P) \vee (P \wedge Q)$$

□ που διαβάζεται ως:

"επιδιώκω την ειρήνη" ΚΑΙ δεν "επιδιώκω την ειρήνη"

'Η "επιδιώκω την ειρήνη" ΚΑΙ "αποφεύγω τον πόλεμο"

❖ Βοηθά στην εύρεση της λογικής τιμής του παραπάνω τύπου.

□ Για την ερμηνεία $I = \{I(P) = T, I(Q) = F\}$ ο τύπος είναι ψευδής.

□ Ο παραπάνω τύπος είναι αληθής μόνο για την ερμηνεία $I = \{I(P) = T, I(Q) = T\}$,

Χρήση των Κανονικών Μορφών

- ❖ Απόδειξη ότι μια συγκεκριμένη λογική έκφραση αποτελεί ταυτολογία.
 - Μετατροπή σε διαζευκτική κανονική μορφή και να απόδειξη ότι μια από τις συζεύξεις αληθεύει πάντα.
- ❖ Εύρεση ερμηνείας που ικανοποιεί ένα τύπο.

Μηχανισμός Εξαγωγής Συμπερασμάτων

- ❖ Έστω ένα σύνολο S καλά σχηματισμένων τύπων σε προτασιακή λογική.
- ❖ Η εξαγωγή συμπερασμάτων αφορά:
 - είτε την δημιουργία όλων των τύπων που λογικά συνεπάγονται από το S ,
 - ή στο να διαπιστωθεί εάν ένας τύπος P λογικά συνεπάγεται από το S , δηλαδή εάν $S \models P$.
- ❖ Η εξαγωγή συμπερασμάτων υλοποιείται είτε
 - με πίνακες αλήθειας ή
 - με την λογική απόδειξη.

Πίνακες Αλήθειας

- ❖ Οι πίνακες αλήθειας (*truth tables*), υπολογίζουν την λογική τιμή ενός τύπου.
- ❖ Ένας τέτοιος πίνακας αποτελείται από 2^N γραμμές όπου N είναι το πλήθος των ατόμων που περιέχονται στο τύπο.

P	Q	$P \vee Q$	$P \rightarrow (P \vee Q)$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	T

- ❖ Απλούστερη μέθοδος εξαγωγής συμπερασμάτων
 - ❑ Ογκωδέστατοι πίνακες αλήθειας.
 - ❑ Π.χ. Η απόδειξη ενός τύπου που περιέχει 15 άτομα απαιτεί ένα πίνακα αλήθειας 2^{15} (32768) γραμμών!

Λογική Απόδειξη

- ❖ Μια απόδειξη (*proof*) είναι μια σειρά από βήματα:
 - Καθένα βήμα είναι η εφαρμογή ενός κανόνα συμπερασμού (*rule of inference*)
 - Απώτερος σκοπός: παραγωγή της αποδεικτέας πρότασης ή την κατάληξη σε άτοπο.

- ❖ Το γεγονός ότι ένας τύπος P μπορεί να αποδειχθεί από ένα αρχικό σύνολο τύπων S , βάσει ενός συνόλου κανόνων συμπερασμού Δ , συμβολίζεται ως $S \vdash_{\Delta} P$.

- ❖ Η χρήση των κανόνων συμπερασμού εξασφαλίζει την ορθότητα των αποτελεσμάτων.

Κανόνες Συμπερασμού

Κανόνας Συμπερασμού		Ονομασία	
(1)	$P_1 \wedge P_2 \wedge \dots \wedge P_N$	$\vdash P_1$	απαλοιφή σύζευξης (and elimination)
(2)	P_1, P_2, \dots, P_N	$\vdash P_1 \wedge P_2 \wedge \dots \wedge P_N$	εισαγωγή συζεύξεων (and introduction)
(3)	P_1	$\vdash P_1 \vee P_2 \vee \dots \vee P_N$	εισαγωγή διαζεύξεων (or introduction)
(4)	$\neg\neg P$	$\vdash P$	απαλοιφή διπλής άρνησης (double negation elimination)
(5)	$P, P \rightarrow Q$	$\vdash Q$	τρόπος του θέτειν (modus ponens)
(6)	$P \vee Q, \neg Q \vee R$	$\vdash P \vee R$	αρχή της ανάλυσης (resolution)

Κανόνες Συμπερασμού

❖ Οι κανόνες συμπερασμού συνήθως γράφονται σαν "κλάσματα".

□ Π.χ. ο κανόνας της απαλοιφής σύζευξης:

$$\frac{P_1 \wedge P_2 \wedge \dots P_N}{P_1}$$

❖ Οι κανόνες εφαρμόζονται στο αρχικό σύνολο προτάσεων μέχρι να παραχθεί η προς απόδειξη πρόταση.

❖ Ο "τρόπος του θέτειν" (*modus ponens*).

□ Εάν είναι γνωστή η αλήθεια των προτάσεων P και $P \rightarrow Q$ μπορούμε να συνάγουμε ότι η πρόταση Q είναι αληθής.

$$\frac{P, P \rightarrow Q}{Q}$$

❖ Από το αρχικό σύνολο προτάσεων:

P : "Ο Νίκος είναι προγραμματιστής"

$P \rightarrow Q$: Εάν "Ο Νίκος είναι προγραμματιστής",
τότε "Ο Νίκος έχει υπολογιστή"

□ χρησιμοποιώντας τον *modus ponens* μπορούμε να συμπεράνουμε ότι:

Q : "Ο Νίκος έχει υπολογιστή"

Διαδικασία Απόδειξης

- ❖ Μια διαδικασία απόδειξης (*proof procedure*) αποτελείται
 - από ένα σύνολο κανόνων συμπερασμού Δ και
 - ένα αλγόριθμο εφαρμογής τους.
- ❖ Δύο σημαντικές έννοιες.
 - Ορθότητα της παραγόμενης γνώσης
 - Ικανότητα της διαδικασίας να εξαγάγει όλα τα δυνατά συμπεράσματα.

Μια αποδεικτική διαδικασία ονομάζεται *ορθή* (*sound*) όταν όλα τα συμπεράσματα που εξάγονται αποτελούν και λογικές συνεπαγωγές του αρχικού συνόλου των τύπων

- Για κάθε P όπου $S \vdash_{\Delta} P$ ισχύει και $S \models P$.

Μια αποδεικτική διαδικασία ονομάζεται *πλήρης* (*complete*) όταν για κάθε τύπο P ο οποίος λογικά συνεπάγεται από ένα σύνολο τύπων S , μπορεί να "κατασκευάσει" μια απόδειξη

- Για κάθε P για το οποίο ισχύει $S \models P$ ισχύει και το $S \vdash_{\Delta} P$.

- ❖ Αυτοματοποίηση της εξαγωγής συμπερασμάτων.
 - Διαδικασία απόδειξης που είναι ορθή, πλήρης αλλά και *αποδοτική* (*efficient*).

Αρχή της Ανάλυσης

❖ Μια διαδικασία ικανή για την αυτοματοποίηση της εξαγωγής συμπερασμάτων βασίζεται στην *αρχή της ανάλυσης (resolution)* (Robinson 1965).

❖ Η *αρχή της ανάλυσης* είναι ο κανόνας συμπερασμού:

$$\frac{P \vee R, \neg P \vee Q}{R \vee Q}$$

❑ P και $\neg P$: συμπληρωματικά ζεύγη (*complementary pairs*)

❑ $R \vee Q$: αναλυθέν (*resolvent*)

❖ Οι προτάσεις θα πρέπει να είναι εκφρασμένες σαν ένα σύνολο διαζεύξεων.

❑ *πρόταση (clause)*: Κάθε διάζευξη αποτελείται από άτομα ή αρνήσεις ατόμων.

❑ Απαιτείται η μετατροπή όλων των προτάσεων στην *συζευκτική μορφή* της λογικής.

❑ Επιτυγχάνεται με την χρήση ισοδυναμιών.

Παράδειγμα Ανάλυσης

❖ Έστω οι προτάσεις:

εάν "έχει ομίχλη" τότε "υπάρχει κίνδυνος" και

εάν "υπάρχει κίνδυνος" τότε "απαιτείται χαμηλή ταχύτητα" και

"έχει ομίχλη"

❖ Σε συμβολική μορφή:

$(\text{"έχει ομίχλη"} \rightarrow \text{"υπάρχει κίνδυνος"}) \wedge$

$(\text{"υπάρχει κίνδυνος"} \rightarrow \text{"απαιτείται χαμηλή ταχύτητα"}) \wedge$

"έχει ομίχλη"

❖ Απαλείφεται το συνδετικό της συνεπαγωγής:

$(\neg \text{"έχει ομίχλη"} \vee \text{"υπάρχει κίνδυνος"}) \wedge$

$(\neg \text{"υπάρχει κίνδυνος"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}) \wedge$

"έχει ομίχλη"

Παράδειγμα Ανάλυσης

❖ Συνήθως χρησιμοποιούμε ένα σύνολο προτάσεων (clauses) παραλείποντας το συνδετικό της σύζευξης.

- (1) $\{\neg\text{"έχει ομίχλη"} \vee \text{"υπάρχει κίνδυνος"}\}$,
- (2) $\neg\text{"υπάρχει κίνδυνος"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}\}$,
- (3) $\text{"έχει ομίχλη"}\}$

❖ Εφαρμογή της αρχής της ανάλυσης για τις πρώτες δύο προτάσεις (1 και 2).

(1) $\neg\text{"έχει ομίχλη"} \vee \text{"υπάρχει κίνδυνος"}$

(2) $\neg\text{"υπάρχει κίνδυνος"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}$
--

(4) $\neg\text{"έχει ομίχλη"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}$

❖ Εφαρμογή της αρχής της ανάλυσης στις προτάσεις (3) και (4).

(3) "έχει ομίχλη"

(4) $\neg\text{"έχει ομίχλη"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}$

(5) $\text{"απαιτείται χαμηλή ταχύτητα"}$

Απόδειξη Βασισμένη στην Αρχή της Ανάλυσης

\neg "υπάρχει κίνδυνος" \vee "απαιτείται χαμηλή ταχύτητα"

\neg "έχει ομίχλη" \vee "υπάρχει κίνδυνος"

"έχει ομίχλη"

\neg "έχει ομίχλη" \vee "απαιτείται χαμηλή ταχύτητα"

"απαιτείται χαμηλή ταχύτητα"

Ορθότητα και Πληρότητα της Αρχής της Ανάλυσης

- ❖ Μια διαδικασία απόδειξης που βασίζεται μόνο στον παραπάνω κανόνα συμπερασμού είναι ορθή.

- ❖ Ο κανόνας της ανάλυσης σε συνδυασμό με την "εις άτοπο απαγωγή" (*refutation ή proof by contradiction*) είναι πλήρης.

- ❖ Απόδειξη αλήθειας μιας πρότασης:
 - εισαγωγή της άρνησης της αποδεικτέας πρότασης
 - προσπάθεια να καταλήξουμε σε άτοπο με εφαρμογή της αρχής της ανάλυσης.

- ❖ Το άτοπο εκφράζεται με την *κενή πρόταση*.
 - Η κενή πρόταση εξάγεται από ένα ζεύγος της μορφής $Q \wedge \neg Q$ και συμβολίζεται με \square

- ❖ Ο κανόνας της ανάλυσης δεν μπορεί να εξαγάγει με *απευθείας απόδειξη* όλους τους δυνατούς τύπους που λογικά συνεπάγονται από την αρχική γνώση.

Παράδειγμα Απαγωγής σε Άτοπο

❖ Αν στο προηγούμενο παράδειγμα απαιτούνταν να αποδειχθεί ότι
"απαιτείται χαμηλή ταχύτητα"

□ από το αρχικό σύνολο προτάσεων

(1) $\{\neg\text{"έχει ομίχλη"} \vee \text{"υπάρχει κίνδυνος"}\}$,

(2) $\neg\text{"υπάρχει κίνδυνος"} \vee \text{"απαιτείται χαμηλή ταχύτητα"}\}$,

(3) $\text{"έχει ομίχλη"}\}$

□ εισάγεται η άρνηση της προς απόδειξη πρότασης

$\neg\text{"απαιτείται χαμηλή ταχύτητα"}$

□ και εφαρμόζεται ο κανόνας συμπερασμού μέχρι να καταλήξει η διαδικασία σε άτοπο.

Απόδειξη βασισμένη στην "εις άτοπο απαγωγή".

\neg "υπάρχει κίνδυνος" \vee "απαιτείται χαμηλή ταχύτητα"

\neg "απαιτείται χαμηλή ταχύτητα"

\neg "έχει ομίχλη" \vee "υπάρχει κίνδυνος"

\neg "υπάρχει κίνδυνος"

"έχει ομίχλη"

\neg "έχει ομίχλη"

□

Πλεονεκτήματα-Μειονεκτήματα Προτασιακής Λογικής

❖ Πλεονεκτήματα της προτασιακής λογικής:

- ❑ Η απλότητα στη σύνταξη
- ❑ Μπορεί να καταλήξει πάντα σε συμπέρασμα (*καταληκτική - decidable*).

❖ Μειονεκτήματα:

- ❑ Έλλειψη γενικότητας.
- ❑ Η προτασιακή λογική υπονοεί ότι ο κόσμος αποτελείται μόνο από γεγονότα τα οποία είναι αληθή ή ψευδή.
- ❑ Καμία δυνατότητα διαχωρισμού και προσπέλασης των οντοτήτων του κόσμου.

Κατηγορηματική Λογική

- ❖ Επέκταση της προτασιακής λογικής.
- ❖ Ο κόσμος περιγράφεται σαν ένα σύνολο αντικειμένων, ιδιοτήτων και σχέσεων.
- ❖ Αντιμετωπίζει το πρόβλημα της μη προσπελασιμότητας των στοιχείων των γεγονότων της προτασιακής λογικής.
 - ❑ Π.χ., η πρόταση " ο τζίμης είναι τίγρης " αναπαριστάται με *τίγρης(τζίμης)*
- ❖ Ύπαρξη μεταβλητών, που αυξάνει σημαντικά την εκφραστική ικανότητά της,
 - ❑ Επιτρέπει την αναπαράσταση "γενικής" γνώσης.
- ❖ Επεκτείνει την προτασιακή λογική εισάγοντας
 - ❑ όρους (terms),
 - ❑ κατηγορήματα (predicates) και
 - ❑ ποσοδείκτες (quantifiers).

Αλφάβητο Κατηγορηματικής Λογικής (1/3)

- ❖ Σταθερές, π.χ. a, b, c, a_1, a_2 , κ.λ.π.
 - Τα ονόματα των σταθερών ξεκινούν με πεζά γράμματα ή αριθμούς.
- ❖ Μεταβλητές, π.χ. X, Y, X_1, X_2, Man κ.λ.π.
 - Αναπαριστώνται από κεφαλαία σύμβολα του λατινικού αλφάβητου, ή τουλάχιστον τα ονόματα τους ξεκινούν με κεφαλαίο γράμμα.
- ❖ Συναρτησιακό σύμβολο, π.χ. $f, g, father_of$ κ.λ.π.
 - Τάξη (*arity*): το πλήθος των ορισμάτων (*arguments*) ή παραμέτρων (*parameters*).
- ❖ Σύμβολα κατηγορημάτων, π.χ. $p, q, color$, κ.λ.π.
 - Κάθε σύμβολο κατηγορήματος έχει μια συγκεκριμένη τάξη.
- ❖ Συνδετικά:
 - Όμοια με εκείνα της προτασιακής λογικής και με την ίδια σημασιολογία.
 - " \wedge " σύζευξη (λογικό "ΚΑΙ"), " \vee " διάζευξη (λογικό "Η"), " \neg " άρνηση, " \rightarrow " συνεπαγωγή ("ΕΑΝ ΤΟΤΕ"), " \leftrightarrow " ισοδυναμία ("ΑΝ ΚΑΙ ΜΟΝΟ ΑΝ").

Αλφάβητο Κατηγορηματικής Λογικής (2/3)

- ❖ Δύο ποσοδείκτες: τον υπαρξιακό ποσοδείκτη " \exists " (existential quantifier) και τον καθολικό ποσοδείκτη " \forall " (universal quantifier).
- ❖ Τρία σύμβολα στίξης: "(", ")", " , " και " , " .
- ❖ Δύο σύμβολα αλήθειας **T** (αληθές) και **F** (ψευδές).
- ❖ Ένας όρος (*term*) της κατηγορηματικής λογικής είναι είτε:
 - ❑ μια σταθερά
 - ❑ μια μεταβλητή
 - ❑ ένας συναρτησιακός όρος (*functional term*) της μορφής $F(t_1, t_2, \dots, t_n)$, όπου F είναι ένα συναρτησιακό σύμβολο τάξης n και τα ορίσματα t_1, t_2, \dots, t_n είναι επίσης όροι.
X, νίκος, πατέραςΤου(νίκου), πατέραςΤου(πατέραςΤου(νίκου))
- ❖ Ένας ατομικός τύπος (atomic formula) έχει την μορφή
$$p(a_1, a_2, \dots, a_n)$$
 - ❑ p είναι ένα σύμβολο κατηγορήματος (ή κατηγορήμα) τάξης n και τα a_1, a_2, \dots, a_n ορίσματα (*arguments*).
 - ❑ Κάθε όρισμα είναι ένας όρος.

Αλφάβητο Κατηγορηματικής Λογικής (3/3)

- ❖ Η σύνδεση προτάσεων για τη δημιουργία *ορθά δομημένων τύπων* γίνεται με τη χρήση συνδετικών.
- ❖ Στην κατηγορηματική λογική οι ορθά δομημένοι τύποι περιέχουν και ποσοδείκτες.
- ❖ Παράδειγμα ορθά δομημένου τύπου:
$$\forall X \text{ φάλαινα}(X) \rightarrow \text{θηλαστικό}(X)$$
- ❖ Για την επεξήγηση του τύπου και την απόδοση λογικής τιμής απαιτείται ορισμός της σημασιολογίας.

Σημασιολογία Κατηγορηματικής Λογικής

- ❖ Αφηρημένος κόσμος (abstract world) ή πεδίου (domain)
 - ❑ Αποτελείται από αντικείμενα και σχέσεις πάνω σε αυτά
 - ❑ Ιδιότητα: σχέση που αφορά μόνο ένα αντικείμενο.
- ❖ Μια *ερμηνεία* αντιστοιχεί τους όρους και ατομικούς τύπους της λογικής στα αντικείμενα και σχέσεις του κόσμου.
- ❖ Η απεικόνιση όρων σε αντικείμενα ονομάζεται *ανάθεση όρων (term assignment)*.
 - ❑ Οι σταθερές αντιστοιχούνται στα αντικείμενα του κόσμου
 - ❑ Οι συναρτησιακοί όροι αναφέρονται σε αντικείμενα, στα οποία δεν δίνουμε ένα συγκεκριμένο όνομα αλλά χρησιμοποιούμε μια περιγραφή για να αναφερθούμε σ' αυτά.
- ❖ Ένας ατομικός τύπος απεικονίζει μια σχέση ανάμεσα σε μια διατεταγμένη πλειάδα (tuple) αντικειμένων
 - ❑ Μπορεί να είναι αληθής ή ψευδής.

Μεταβλητές και Ποσοδείκτες

Στην κατηγορηματική λογική πρώτης τάξης (*first order predicate logic*) οι μεταβλητές αναφέρονται μόνο σε αντικείμενα και όχι σε συναρτησιακά σύμβολα ή κατηγορήματα.

❖ Επιβάλλεται η ποσοτικοποίηση των μεταβλητών από έναν από τους ποσοδείκτες.

άνθρωπος (X) \rightarrow θνητός (X)

άνθρωπος (X) \wedge μαθηματικός (X)

❖ Τι ακριβώς αναπαριστούν οι παραπάνω τύποι;

❖ Η αποσαφήνιση της σημασίας των παραπάνω εκφράσεων απαιτεί την εισαγωγή κατάλληλων ποσοδεικτών.

❖ Ο *υπαρξιακός ποσοδείκτης* " \exists " (*existential quantifier*).

□ $(\exists X) (\varphi (X))$ προφέρεται "υπάρχει X , τέτοιο ώστε ο τύπος $\varphi (X)$ να αληθής"

❖ Ο *καθολικός ποσοδείκτης* " \forall " (*universal quantifier*).

□ $(\forall X) (\varphi (X))$ προφέρεται "για κάθε X , ο $\varphi (X)$ είναι αληθής"

❖ Έτσι μια σωστότερη αναπαράσταση της παραπάνω γνώσης είναι:

$(\forall X) (\text{άνθρωπος}(X) \rightarrow \text{θνητός}(X))$

"όλοι οι άνθρωποι είναι θνητοί"

$(\exists X) (\text{άνθρωπος}(X) \wedge \text{μαθηματικός}(X))$

"κάποιος άνθρωπος είναι μαθηματικός"

Εμβέλεια Ποσοδεικτών

- ❖ Σε έναν τύπο $(\exists \mathbf{X}) (\varphi(\mathbf{X}))$ ή $(\forall \mathbf{X}) (\varphi(\mathbf{X}))$, το $\varphi(\mathbf{X})$ ονομάζεται *εμβέλεια (scope)* των $\exists \mathbf{X}$ και $\forall \mathbf{X}$ αντίστοιχα.
- Η εμβέλεια του $\forall \mathbf{X}$ στον τύπο $(\forall \mathbf{X}) (\text{άνθρωπος}(\mathbf{X}) \rightarrow \text{θνητός}(\mathbf{X}))$ είναι ο τύπος $\text{άνθρωπος}(\mathbf{X}) \rightarrow \text{θνητός}(\mathbf{X})$.
- ❖ Μια εμφάνιση (*occurrence*) κάποιας μεταβλητής, μπορεί να είναι:
 - δεσμευμένη (*bound*).
 - ελεύθερη (*free*).
 - Πχ, στον τύπο $(\forall \mathbf{X}) (\varphi(\mathbf{X}, \mathbf{Y}))$
 - η εμφάνιση της μεταβλητής \mathbf{X} είναι *δεσμευμένη*
 - η εμφάνιση της μεταβλητής \mathbf{Y} είναι *ελεύθερη*.
- ❖ Ένας τύπος που στερείται ελεύθερων μεταβλητών ονομάζεται *κλειστός τύπος (closed formula)*.
- ❖ *Βασικός όρος (ή τύπος) (ground term)*, είναι ένας όρος (ή τύπος) που δεν περιέχει καμία μεταβλητή.
 - Βασικοί όροι: *επάγγελμα(προγραμματιστής)* και *φορολογούμενος(νίκος, επάγγελμα(προγραμματιστής))*

Σειρά Ποσοδεικτών

- ❖ Η ύπαρξη πολλών ποσοτικοποιημένων μεταβλητών απαιτεί περισσότερη προσοχή.
- Η σειρά των μεταβλητών παίζει σημαντικό ρόλο.

❖ Για παράδειγμα.

$(\forall X) ((\exists Y) (\text{δεσμός}(X, Y)))$

$(\exists Y) ((\forall X) (\text{δεσμός}(X, Y)))$

❖

❖ Ο πρώτος σημαίνει "για κάθε κόμβο X του γράφου, δηλαδή για ένα αντικείμενο του πεδίου, υπάρχει ένας κόμβος Y έτσι ώστε να ισχύει η σχέση δεσμός(X, Y)".

❖ Ο δεύτερος σημαίνει ότι "υπάρχει τουλάχιστον ένας κόμβος Y τέτοιος ώστε για κάθε κόμβο X του γράφου να ισχύει η σχέση δεσμός(X, Y)".

Αντικατάσταση και Ενοποίηση

- ❖ Η αντικατάσταση (*substitution*) αφορά την αντικατάσταση των μεταβλητών από κάποιους όρους.
- ❖ Παριστάνεται με $\{X_i/t_i\}$ όπου X_i η μεταβλητή και t_i ο όρος.
 - Π.χ. η αντικατάσταση $\{X/\text{φάλαινα}\}$ στον τύπο:
είναι $(X, \text{θηλαστικό})$
 - θα δώσει τον τύπο:
είναι $(\text{φάλαινα}, \text{θηλαστικό})$

Ενοποίηση (unification) είναι η διαδικασία κατά την οποία δύο εκφράσεις γίνονται συντακτικά όμοιες με την χρήση αντικαταστάσεων.

- Π.χ. οι ακόλουθες προτάσεις:
είναι $(\text{λιοντάρι}, \text{θηλαστικό}, X)$ είναι $(\text{λιοντάρι}, Y, \text{σαρκοβόρο})$
- ενοποιούνται με την αντικατάσταση $\theta = \{X/\text{σαρκοβόρο}, Y/\text{θηλαστικό}\}$.

Ενοποιητής

- ❖ Για δύο εκφράσεις φ_1 και φ_2 , ο *ενοποιητής* (*unifier*) τους, είναι μια αντικατάσταση θ τέτοια ώστε η έκφραση $\varphi_1\theta$ να είναι συντακτικά όμοια με την $\varphi_2\theta$.
- ❖ Οι εκφράσεις φ_1 και φ_2 ονομάζονται *ενοποιήσιμες* (*unifiable*).
- ❖ Ο *γενικότερος ενοποιητής* (*mgu-most general unifier*) ενοποιεί τις εκφράσεις με τις λιγότερες δυνατές αντικαταστάσεις.
- ❖ Εύρεση του *γενικότερου ενοποιητή*:
 - Δύο σταθερές ενοποιούνται αν και μόνο αν είναι ίδιες.
 - Μια μεταβλητή ενοποιείται με οποιοδήποτε όρο.
 - Δύο συναρτησιακοί όροι ενοποιούνται αν έχουν το ίδιο συναρτησιακό σύμβολο, την ίδια τάξη (αριθμό ορισμάτων) και αν κάθε όρισμα του πρώτου μπορεί να ενοποιηθεί με το αντίστοιχο σε θέση όρισμα του δεύτερου όρου.

Δυο ατομικοί τύποι ενοποιούνται αν έχουν το ίδιο κατηγορημα, την ίδια τάξη (αριθμό ορισμάτων) και αν κάθε όρισμα του πρώτου μπορεί να ενοποιηθεί με το αντίστοιχο σε θέση όρισμα του δεύτερου ατομικού τύπου.

Παράδειγμα Αναπαράστασης Γνώσης

- ❖ Γνώση για τα χαρακτηριστικά διαφόρων ειδών ζώων:
 - ❑ Κάθε ζώο το οποίο έχει τρίχωμα ή παράγει γάλα είναι θηλαστικό.
 - ❑ Κάθε ζώο που έχει φτερά και γεννάει αυγά είναι πουλί.
 - ❑ Κάθε θηλαστικό που τρέφεται με κρέας ή έχει κοφτερά δόντια είναι σαρκοβόρο.
 - ❑ Κάθε σαρκοβόρο με χρώμα καφέ-πορτοκαλί που έχει μαύρες ρίγες είναι τίγρης.
 - ❑ Κάθε σαρκοβόρο με χρώμα καφέ-πορτοκαλί που έχει μαύρες βούλες είναι τσιτάχ.
 - ❑ Κάθε πουλί το οποίο δεν πετά και κολυμπά είναι πιγκουΐνος.

❖ Οι προτάσεις αναπαριστώνται σε κατηγορηματική λογική ως:

$\forall X (\text{έχει}(X, \text{τρίχωμα}) \vee \text{παράγει}(X, \text{γάλα})) \rightarrow \text{είναι}(X, \text{θηλαστικό})$

$\forall X (\text{έχει}(X, \text{φτερά}) \wedge \text{γεννάει}(X, \text{αυγά})) \rightarrow \text{είναι}(X, \text{πουλί})$

$\forall X (\text{είδος}(X, \text{θηλαστικό}) \wedge ((\text{τρέφεται}(X, \text{κρέας}) \vee \text{έχει}(X, \text{δόντια(κοφτερά)}))) \rightarrow \text{είναι}(X, \text{σαρκοβόρο})$

$\forall X (\text{είναι}(X, \text{σαρκοβόρο}) \wedge \text{χρώμα}(X, \text{καφέ-πορτοκαλί}) \wedge \text{έχει}(X, \text{ρίγες(μαύρες)}) \rightarrow \text{είναι}(X, \text{τίγρης})$

$\forall X (\text{είναι}(X, \text{σαρκοβόρο}) \wedge \text{χρώμα}(X, \text{καφέ-πορτοκαλί}) \wedge \text{έχει}(X, \text{βούλες(μαύρες)}) \rightarrow \text{είναι}(X, \text{τσιτάχ}) .$

$\forall X (\text{είναι}(X, \text{πουλί}) \wedge (\neg \text{πετά}(X)) \wedge \text{κολυμπά}(X)) \rightarrow \text{είναι}(X, \text{πιγκουΐνος}) .$

❖ Πως εξάγεται νέα γνώση (νέες προτάσεις);

Ισοδυναμίες

- ❖ Υπάρχει σύνολο ισοδυναμιών για το μετασχηματισμό των τύπων της λογικής.
- ❖ Δύο "κατηγορίες" ισοδυναμιών":
 - ❑ Κοινές με την προτασιακή λογική
 - ❑ Εκείνες που αφορούν ποσοδείκτες

Από την Προτασιακή Λογική

Ισοδυναμία		Ονομασία
(1)	$p \Leftrightarrow \neg\neg p$	νόμος της διπλής άρνησης
(2)	$(\neg p \vee \neg q) \Leftrightarrow \neg(p \wedge q)$	νόμος De Morgan
(3)	$(\neg p \wedge \neg q) \Leftrightarrow \neg(p \vee q)$	νόμος De Morgan
(4)	$(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$	επιμερισμός ως προς την σύζευξη
(5)	$(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$	επιμερισμός ως προς την διάζευξη
(6)	$(p \rightarrow q) \Leftrightarrow \neg p \vee q$	
(7)	$(p \leftrightarrow q) \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$	

Ισοδυναμίες Για Ποσοδείκτες

(8)	$\forall X(p(X)) \Leftrightarrow \neg \exists X(\neg p(X))$	
(9)	$\exists X(p(X)) \Leftrightarrow \neg \forall X(\neg p(X))$	
(10)	$\forall X(p(X)) \vee q \Leftrightarrow \forall X(p(X) \vee q)$	όπου το q δεν περιέχει ελεύθερες εμφανίσεις της μεταβλητής X.
(11)	$\forall X(p(X)) \wedge q \Leftrightarrow \forall X(p(X) \wedge q)$	
(12)	$\exists X(p(X)) \vee q \Leftrightarrow \exists X(p(X) \vee q)$	
(13)	$\exists X(p(X)) \wedge q \Leftrightarrow \exists X(p(X) \wedge q)$	
(14)	$\forall X(p(X)) \Leftrightarrow \forall Y(p(Y))$	
(15)	$\exists X(p(X)) \Leftrightarrow \exists Y(p(Y))$	μετονομασία μεταβλητών
(16)	$\forall X(p(X)) \wedge \forall X(q(X)) \Leftrightarrow \forall X(p(X) \wedge q(X))$	
(17)	$\exists X(p(X)) \vee \exists X(q(X)) \Leftrightarrow \exists X(p(X) \vee q(X))$	

Παρατηρήσεις πάνω στις ισοδυναμίες

❖ Η μετονομασία δεσμευμένων μεταβλητών σε ένα τύπο και στην εμφάνιση του αντίστοιχου ποσοδείκτη, διατηρεί την ισοδυναμία (ισοδυναμίες (14) και (15)).

$\forall X$ (έχει (X, τρίχωμα) \vee παράγει (X, γάλα)) \rightarrow είναι (X, θηλαστικό)

$\forall Z$ (έχει (Z, τρίχωμα) \vee παράγει (Z, γάλα)) \rightarrow είναι (Z, θηλαστικό)

❖ Η σύζευξη καθολικά ποσοτικοποιημένων τύπων είναι ισοδύναμη με την καθολικά ποσοτικοποιημένη σύζευξή τους.

□ $\forall X(p(X)) \wedge \forall X(q(X)) \Leftrightarrow \forall X(p(X) \wedge q(X))$

❖ Η διάζευξη υπαρξιακά ποσοτικοποιημένων τύπων είναι ισοδύναμη με την υπαρξιακά ποσοτικοποιημένη διάζευξή τους.

□ $\exists X(p(X)) \vee \exists X(q(X)) \Leftrightarrow \exists X(p(X) \vee q(X))$

❖ ΔΕΝ ισχύουν οι ισοδυναμίες

□ $\forall X(p(X)) \vee \forall X(q(X)) \Leftrightarrow \forall X(p(X) \vee q(X))$

□ $\exists X(p(X)) \wedge \exists X(q(X)) \Leftrightarrow \exists X(p(X) \wedge q(X))$

Προσημασμένη Συζευκτική Κανονική Μορφή

❖ Ὑπαρξη τύπων που αν και φαινομενικά διαφορετικοί, είναι λογικά ισοδύναμοι.

$$\neg((\exists X) (p(X) \rightarrow q(X))) \\ (\forall X) (p(X) \wedge \neg q(X))$$

❖ Αναγωγή σε μια κανονική μορφή.

❖ Προσημασμένη συζευκτική κανονική μορφή (*prenex conjunctive normal form*):

$$\forall X \exists Y (p(X) \wedge \neg q(X) \wedge (p(X) \vee \neg p(X) \vee q(X, Y)) \wedge \dots \wedge (r(X, Y) \vee s(X)))$$

❖ Τα βασικά δομικά στοιχεία:

□ τα λεκτικά στοιχεία (*literals*) (ατομικός τύπος ή η άρνηση ενός ατομικού τύπου)

□ οι προτάσεις (*clauses*) (πεπερασμένη διάζευξη (*disjunction*) κανενός ή περισσότερων λεκτικών στοιχείων).

□ Π.χ. η έκφραση $p(X) \vee \neg p(X) \vee q(X, Y)$ είναι μια πρόταση.

❖ Η Κενή πρόταση (*empty clause*) αναπαρίσταται με το σύμβολο \square .

❖ Ένας τύπος (*formula*) αποτελείται από μια σύζευξη προτάσεων προσημασμένης από υπαρξιακούς και καθολικούς ποσοδείκτες.

$$\forall X (\neg \text{έχει}(X, \text{τρίχωμα}) \vee \neg \text{παράγει}(X, \text{γάλα}) \vee \text{είναι}(X, \text{θηλαστικό})) . \\ \forall X ((\neg \text{έχει}(X, \text{φτερά}) \vee \text{είναι}(X, \text{πουλί})) \\ \wedge (\neg \text{γεννάει}(X, \text{αυγά}) \vee \text{είναι}(X, \text{πουλί})))$$

Διαδικασία Αναγωγής σε Κανονική Μορφή

- ❖ Οποιοσδήποτε τύπος της κατηγορηματικής λογικής μπορεί να αναχθεί σε ένα ισοδύναμο τύπο της προσημασμένης συζευκτικής κανονικής μορφής της λογικής.

- ❖ Η διαδικασία περιλαμβάνει τα ακόλουθα βήματα:
 - Απαλοιφή των συνδετικών της ισοδυναμίας και συνεπαγωγής (ισοδυναμίες (6), (7))
 - Μετονομασία των μεταβλητών έτσι ώστε δύο μεταβλητές που ποσοτικοποιούνται από διαφορετικούς ποσοδείκτες να μην έχουν το ίδιο όνομα. (ισοδυναμίες (14), (15))
 - Μετατροπή των τύπων έτσι ώστε το συνδετικό της άρνησης να εφαρμόζεται μόνο σε ατομικούς τύπους (ισοδυναμίες (1), (2), (3), (8) και (9))
 - Μεταφορά των ποσοδεικτών με αναδρομική εφαρμογή των ισοδυναμιών (10)–(13)
 - Εφαρμογή των ισοδυναμιών επιμερισμού ως προς την σύζευξη και διάζευξη έτσι ώστε ο τελικός τύπος να αποτελείται από συζεύξεις προτάσεων (ισοδυναμίες (4), (5)).

Παράδειγμα Αναγωγής σε Κανονική Μορφή

$$\forall X (\text{βλάβη}(X) \rightarrow \exists Y (\text{σύμπτωμα}(X, Y))) \wedge \neg(\exists Y (\text{βλάβη}(Y) \wedge \text{μηχάνημα}(\text{λειτουργεί})))$$

- ❖ Απαλοιφή του συνδετικού της ισοδυναμίας (ισοδυναμία (6)):

$$\forall X (\neg\text{βλάβη}(X) \vee \exists Y (\text{σύμπτωμα}(X, Y))) \wedge \neg(\exists Y (\text{βλάβη}(Y) \wedge \text{μηχάνημα}(\text{λειτουργεί})))$$

- ❖ Επειδή η μεταβλητή Y εμφανίζεται ποσοτικοποιημένη από δύο διαφορετικούς ποσοδείκτες, η δεύτερη της εμφάνιση μετονομάζεται σε Z

$$\forall X (\neg\text{βλάβη}(X) \vee \exists Y (\text{σύμπτωμα}(X, Y))) \wedge \neg(\exists Z (\text{βλάβη}(Z) \wedge \text{μηχάνημα}(\text{λειτουργεί})))$$

- ❖ Εφαρμογή των ισοδυναμιών DeMorgan και (9) (άρνηση μόνο σε τύπους):

$$\forall X (\neg\text{βλάβη}(X) \vee \exists Y (\text{σύμπτωμα}(X, Y))) \wedge \forall Z (\neg\text{βλάβη}(Z) \vee \neg\text{μηχάνημα}(\text{λειτουργεί}))$$

- ❖ Εφαρμογή των ισοδυναμιών (10) και (12) (ομαδοποίηση των λεκτικών):

$$\forall X \exists Y \forall Z ((\neg\text{βλάβη}(X) \vee \text{σύμπτωμα}(X, Y)) \wedge (\neg\text{βλάβη}(Z) \vee \neg\text{μηχάνημα}(\text{λειτουργεί})))$$

Κανονική Μορφή κατά Skolem

❖ Είναι δυνατό να υπάρξει κάποια κανονική μορφή στην οποία να εξαλείφονται πλήρως οι ποσοδείκτες;

❖ *Κανονική μορφή κατά Skolem,*

□ οι υπαρξιακά ποσοτικοποιημένες εμφανίσεις μεταβλητών αντικαθίστανται από σταθερές ή συναρτήσεις καθολικά ποσοτικοποιημένων μεταβλητών.

1) Έστω $\exists X_i$ η πρώτη από αριστερά υπαρξιακά ποσοτικοποιημένη μεταβλητή στον τύπο και $\forall X_1 \dots \forall X_{i-1}$ οι καθολικά ποσοτικοποιημένες μεταβλητές του τύπου, μέσα στην εμβέλεια των οποίων βρίσκεται το $\exists X_i$, δηλαδή που βρίσκονται στα "αριστερά του $\exists X_i$, τότε

a) Αν το πλήθος των καθολικών μεταβλητών $(X_1 \dots X_{i-1})$ είναι μηδέν, δηλαδή δεν υπάρχουν καθολικά ποσοτικοποιημένες μεταβλητές στα αριστερά της X_i , τότε κάθε εμφάνιση της X_i στο τύπο αντικαθίσταται από μια νέα σταθερά Skolem sk_{X_i} (Skolem constant).

b) Αν το πλήθος των καθολικών μεταβλητών είναι μεγαλύτερο του μηδενός, τότε κάθε εμφάνιση της μεταβλητής X_i στον τύπο αντικαθίσταται από μια νέα συνάρτηση Skolem (Skolem function) στις μεταβλητές $X_1 \dots X_{i-1}$, $sk_funcX_i(X_1, \dots, X_{i-1})$.

c) Διάγραψε τον υπαρξιακό ποσοδείκτη $\exists X_i$ από τον τύπο.

2) Αν υπάρχουν άλλοι υπαρξιακοί ποσοδείκτες στον τύπο, τότε πήγαινε στο βήμα 1.

3) Διάγραψε όλους τους καθολικούς ποσοδείκτες.

Παράδειγμα Κανονικής Μορφής κατά Skolem

$$\forall X \exists Y \forall Z ((\neg \text{βλάβη}(X) \vee \text{σύμπτωμα}(X, Y)) \wedge (\neg \text{βλάβη}(Z) \vee \neg \text{μηχάνημα}(\text{λειτουργεί})))$$

❖ Υπάρχει μόνο ένας υπαρξιακός ποσοδείκτης ($\exists Y$) μέσα στην εμβέλεια του καθολικού ποσοδείκτη $\forall X$.

❖ Ο ποσοδείκτης αντικαθίσταται από την συνάρτηση Skolem $\text{sk_function}_Y(X)$:

$$\forall X \forall Z ((\neg \text{βλάβη}(X) \vee \text{σύμπτωμα}(X, \text{sk_func}_Y(X))) \\ \wedge (\neg \text{βλάβη}(Z) \vee \neg \text{μηχάνημα}(\text{λειτουργεί})))$$

❖ Διαγράφονται όλοι οι καθολικοί ποσοδείκτες από τον παραπάνω τύπο:

$$(\neg \text{βλάβη}(X) \vee \text{σύμπτωμα}(X, \text{sk_func}_Y(X))) \\ \wedge (\neg \text{βλάβη}(Z) \vee \neg \text{μηχάνημα}(\text{λειτουργεί}))$$

❖ Η συνάρτηση Skolem δεν εξαρτάται από την μεταβλητή Z αλλά μόνο από την X .

❖ Ο νέος τύπος που προκύπτει είναι *ασθενώς ισοδύναμος* (*weakly equivalent*) με τον αρχικό τύπο.

❑ Διατηρείται η μη-ικανοποιησιμότητα (unsatisfiability).

❑ Πληρότητα αποδεικτικών διαδικασιών βασισμένων στην *εις άτοπο απαγωγή*.

Προτασιακή Μορφή της Κατηγορηματικής Λογικής

❖ Μετατροπή ενός τύπου κανονικής μορφής σε ένα σύνολο προτάσεων- προτασιακή μορφή της λογικής (clausal form).

❖ Ο μετασχηματισμός βασίζεται στον κανόνα της απαλοιφής σύζευξης: $\frac{p \wedge q}{p}$, $\frac{p \wedge q}{q}$

□ Ένας τύπος της μορφής $p \wedge q \wedge r \wedge s$ μετατρέπεται στο σύνολο των τύπων $\{p, q, r, s\}$

❖ Το σύνολο των τύπων που προκύπτει αποτελείται από προτάσεις.

❖ Για παράδειγμα ο τύπος

$$\left(\neg \text{βλάβη}(X) \vee \text{σύμπτωμα}(X, \text{sk_func}_Y(X)) \right) \wedge \\ \left(\neg \text{βλάβη}(Z) \vee \neg \text{μηχάνημα}(\text{λειτουργεί}) \right)$$

□ μετατρέπεται στις προτάσεις:

$$\neg \text{βλάβη}(X) \vee \text{σύμπτωμα}(X, \text{sk_func}_Y(X)) \\ \neg \text{βλάβη}(Z) \vee \neg \text{μηχάνημα}(\text{λειτουργεί})$$

Μορφή Kowalski

❖ Όλες οι προτάσεις εκφράζονται σαν λογικές ισοδυναμίες της μορφής

$$\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n \rightarrow \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m$$

- ❑ Οι ατομικοί τύποι \mathbf{r}_i είναι σε διάζευξη, ενώ οι \mathbf{q}_j σε σύζευξη.
- ❑ Τα \mathbf{r}_i αποτελούν τα *συμπεράσματα* της πρότασης, ενώ τα \mathbf{q}_j τις *υποθέσεις* της.
- ❑ Τόσο τα συμπεράσματα όσο και οι υποθέσεις δεν περιέχουν αρνήσεις ατομικών τύπων.
- ❖ Η διαδικασία μετατροπής μιας πρότασης σε μορφή Kowalski είναι εξαιρετικά απλή.

❑ Για παράδειγμα έστω η πρόταση

$$\mathbf{p} \vee \neg \mathbf{q} \vee \mathbf{r} \vee \neg \mathbf{s} \vee \mathbf{t}$$

❑ Το πρώτο βήμα αφορά την συγκέντρωση όλων των ατομικών τύπων σε άρνηση στο αριστερό μέρος της πρότασης, με εφαρμογή της ισοδυναμίας $\mathbf{p} \vee \mathbf{q} \Leftrightarrow \mathbf{q} \vee \mathbf{p}$:

$$\neg \mathbf{q} \vee \neg \mathbf{s} \vee \mathbf{p} \vee \mathbf{r} \vee \mathbf{t}$$

❑ Εφαρμογή του νόμου DeMorgan $\neg \mathbf{p} \vee \neg \mathbf{q} \Leftrightarrow \neg (\mathbf{p} \wedge \mathbf{q})$

$$\neg (\mathbf{q} \wedge \mathbf{s}) \vee \mathbf{p} \vee \mathbf{r} \vee \mathbf{t}$$

❑ Εφαρμογή της ισοδυναμίας $(\mathbf{p} \rightarrow \mathbf{q}) \Leftrightarrow \neg \mathbf{p} \vee \mathbf{q}$

$$\mathbf{q} \wedge \mathbf{s} \rightarrow \mathbf{p} \vee \mathbf{r} \vee \mathbf{t}$$

❑ Αντικατάσταση των συμβόλων της σύζευξης και της διάζευξης με το σύμβολο ",".

$q, s \rightarrow p, r, t$

Παράδειγμα Μορφής Kowalski

- ❖ Για παράδειγμα οι προτάσεις:
 - βλάβη (X) \vee σύμπτωμα (X, θόρυβος) \vee ένταση (θόρυβος, μεγάλη)
 - βλάβη (W) \vee –εξάρτημα (Z, W, δεν_λειτουργεί) \vee αντικατάσταση (Z)
- ❖ στην μορφή Kowalski
 - βλάβη (X) \rightarrow σύμπτωμα (X, θόρυβος) , ένταση (θόρυβος, μεγάλη)
 - βλάβη (W) , εξάρτημα (Z, W, δεν_λειτουργεί) \rightarrow αντικατάσταση (Z)
- ❖ Περισσότερο αναγνώσιμη μορφή.

Περιπτώσεις Πρότασεων Kowalski

$$q_1, q_2, \dots, q_n \rightarrow r_1, r_2, \dots, r_m$$

❖ Αν $m > 0$ και $n > 0$, τότε η πρόταση ερμηνεύεται σαν
ισχύει r_1 ή r_2 ...ή r_m εάν q_1 και q_2 ...και q_n

□ προτάσεις Horn (Horn clauses).

□ Επιτρέπεται μόνο ένας ατομικός τύπος στο συμπέρασμα, είναι δηλαδή της μορφής:

$$q_1, q_2, \dots, q_n \rightarrow r$$

Κανόνας

❖ Αν $m=0$, τότε οι υποθέσεις καταλήγουν σε αναληθή συμπέρασμα:

$$q_1, q_2, \dots, q_n \rightarrow$$

Στόχος, Ερώτηση

❖ Αν $n=0$, τότε αναπαριστάται μια πρόταση χωρίς υπόθεση.

$$\rightarrow r_1, r_2, \dots, r_m$$

Γεγονότα

❖ Αν $m=0$ και $n=0$, τότε αναπαριστάται μια πρόταση πάντα αναληθή και συμβολίζεται με την κενή πρόταση □.

Μηχανισμός Εξαγωγής Συμπερασμάτων

❖ Ο βασικός μηχανισμός εξαγωγής συμπερασμάτων στην κατηγορηματική λογική είναι η *απόδειξη*.

❖ Υπάρχει ένα πλήθος κανόνων συμπερασμού.

□ Αυτοί που ισχύουν στην προτασιακή λογική,

□ Επιπλέον υπάρχουν δύο κανόνες που αφορούν προτάσεις που περιέχουν ποσοτικοποιημένες μεταβλητές.

□ Όλοι οι κανόνες βασίζονται στην έννοια της αντικατάστασης μεταβλητών.

❖ Για παράδειγμα έστω ότι η βάση γνώσης περιέχει τις προτάσεις

$$(\forall X) (\text{άνθρωπος}(X) \rightarrow \text{θνητός}(X)) \quad (1)$$

$$\text{άνθρωπος}(\nu \text{ίκος}) \quad (2)$$

□ Σύμφωνα με τον κανόνα συμπερασμού (7) και την αντικατάσταση $\theta = \{X/\nu \text{ίκος}\}$ είναι δυνατό να εξαχθεί το συμπέρασμα:

$$\text{άνθρωπος}(\nu \text{ίκος}) \rightarrow \text{θνητός}(\nu \text{ίκος}) \quad (3)$$

□ Με εφαρμογή του κανόνα "*τρόπος του θέτειν*" (*modus ponens*) στις προτάσεις (2) και (3), εξάγεται η πρόταση

$$\text{θνητός}(\nu \text{ίκος}) \quad (4)$$

Κανόνες Συμπερασμού της Κατηγορηματικής Λογικής

Κανόνας Συμπερασμού			Ονομασία
(1)	$p_1 \wedge p_2 \wedge \dots \wedge p_n$	$\vdash p_i$	απαλοιφή σύζευξης (and elimination)
(2)	p_1, p_2, \dots, p_n	$\vdash p_1 \wedge p_2 \wedge \dots \wedge p_n$	εισαγωγή συζεύξεων (and introduction)
(3)	p_i	$\vdash p_1 \vee p_2 \vee \dots \vee p_n$	εισαγωγή διαζεύξεων (or introduction)
(4)	$\neg\neg p$	$\vdash p$	απαλοιφή διπλής άρνησης (double negation elimination)
(5)	$p, p \rightarrow q$	$\vdash q$	τρόπος του θέτειν (modus ponens)
(6)	$p \vee q, \neg q \vee r$	$\vdash p \vee r$	αρχή της ανάλυσης (resolution)
(7)	$\forall X p(X)$	$\vdash p(a) \theta=\{X/a\}$	απαλοιφή καθολικού ποσοδείκτη (universal elimination)
(8)	$P(a)$	$\vdash \exists X p(X) \theta=\{X/a\}$	εισαγωγή υπαρξιακού ποσοδείκτη (existential introduction)

Γενικευμένος Τρόπος του Θέτειν

❖ "Γενικευμένος τρόπος του θέτειν" - ΓΤΘ (*generalized modus ponens*):

$$\frac{p'_1, p'_2, \dots, p'_n, p_1 \wedge p_2 \dots \wedge p_n \rightarrow q}{\theta q}$$

□ όπου θ το σύνολο των αντικαταστάσεων οι οποίες κάνουν τα p'_i και p_i συντακτικά όμοια.

❖ Η εξαγωγή του συμπεράσματος γίνεται σε ένα βήμα:

$(\forall X) (\text{άνθρωπος}(X) \rightarrow \text{θνητός}(X))$
$\text{άνθρωπος}(\text{νίκος})$
$\text{θνητός}(\text{νίκος}) \quad (\text{με } \theta = \{X/\text{νίκος}\})$

❖ Μια αποδεικτική διαδικασία που χρησιμοποιεί ως μοναδικό κανόνα συμπερασμού τον ΓΤΘ είναι *ορθή* αλλά στην γενική περίπτωση *δεν είναι πλήρης*.

□ Μια τέτοια διαδικασία είναι πλήρης μόνο αν η βάση γνώσης αποτελείται από προτάσεις Horn.

Η Αρχή της Ανάλυσης στη Κατηγορηματική Λογική

Η αρχή της ανάλυσης (*resolution*) είναι ο μοναδικός κανόνας που απαιτείται για την εξαγωγή όλων των σωστών συμπερασμάτων σε μια αποδεικτική διαδικασία που χρησιμοποιεί τη μέθοδο της "εις άτοπο απαγωγής" (*refutation*).

□ Η διαδικασία απόδειξης είναι ορθή και πλήρης.

❖ Στην απλή περίπτωση περιλαμβάνει προτάσεις οι οποίες περιέχουν το πολύ δύο λεκτικά στοιχεία (*literals*):

$$\frac{p \vee \neg q, z \vee q'}{\theta(p \vee z)}$$

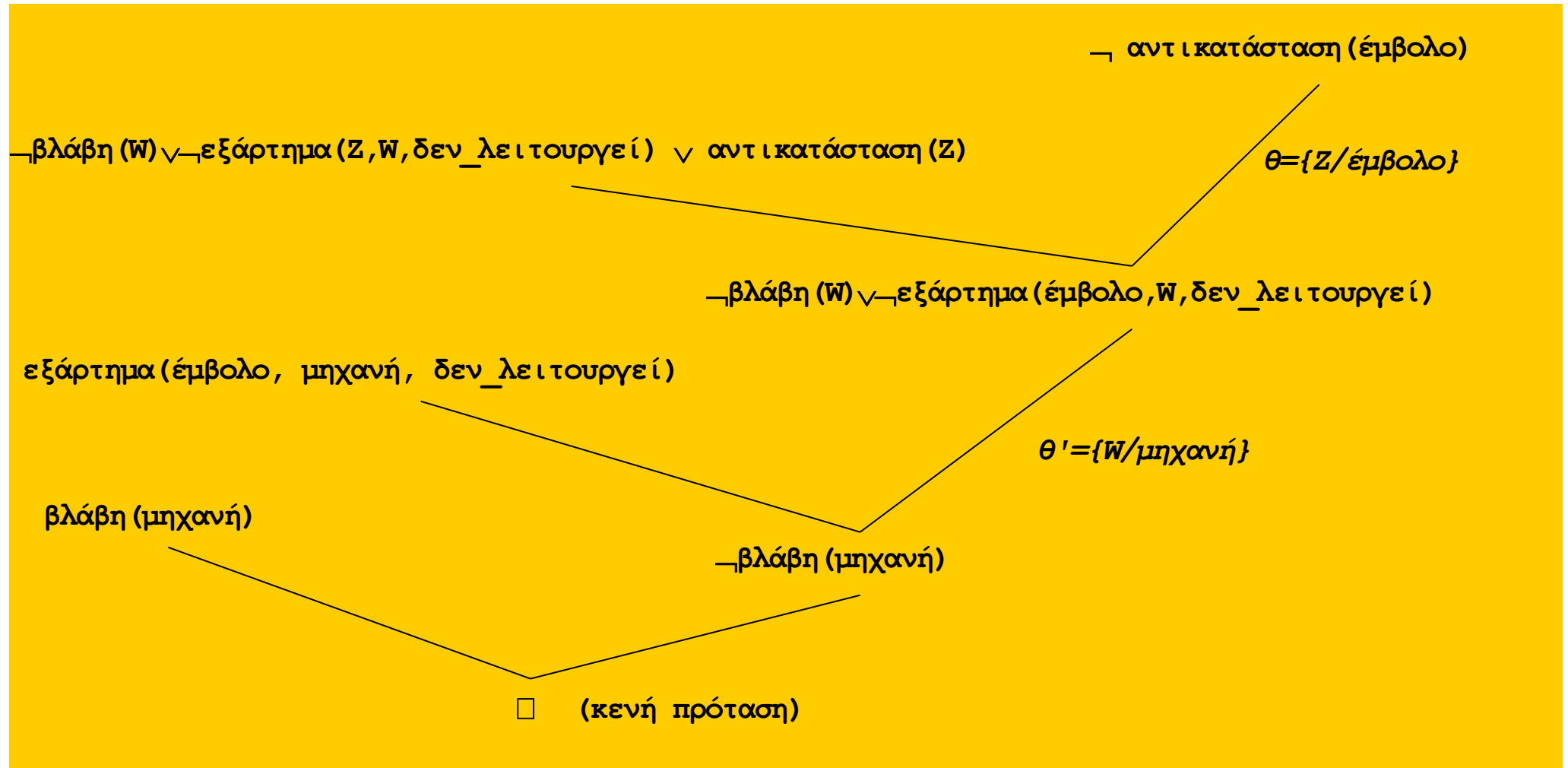
□ Τα λεκτικά στοιχεία q' και $\neg q$ ονομάζονται συμπληρωματικά ζεύγη

□ Οι αντικαταστάσεις μεταβλητών που προκύπτουν εφαρμόζονται στο αναλυθέν (*resolvent*) $p \vee z$.

Διαδικασία Απόδειξης

- ❖ Η διαδικασία απόδειξης περιλαμβάνει την
 - εισαγωγή της άρνησης της προς απόδειξη πρότασης στο αρχικό σύνολο προτάσεων
 - εφαρμογή του κανόνα της ανάλυσης μέχρι το σύστημα να εξαγάγει την κενή πρόταση (άτοπο).
- ❖ Π.χ. έστω ότι θέλουμε να διερευνήσουμε αν το έμβολο μιας μηχανής χρειάζεται αντικατάσταση:
 - βλάβη (μηχανή)
 - σύμπτωμα (έμβολο, θόρυβος)
 - μέρος (έμβολο, μηχανή)
 - εξάρτημα (έμβολο, μηχανή, δεν_λειτουργεί)
 - \neg βλάβη (W) \vee \neg εξάρτημα (Z, W, δεν_λειτουργεί) \vee αντικατάσταση (Z)
 - \neg βλάβη (X) \vee \neg σύμπτωμα (R, θόρυβος) \vee \neg μέρος (R, X) \vee αντικατάσταση (R)
- Εισάγεται η άρνηση της πρότασης αντικατάσταση (έμβολο) και εφαρμόζεται διαδοχικά ο κανόνας της ανάλυσης.
- ❖ Οι προτάσεις πρέπει να είναι σε μορφή διαζεύξεων (προτασιακή μορφή της κατηγορηματικής λογικής).

Απόδειξη βασισμένη στην αρχή της ανάλυσης στην Κατηγορηματική Λογική



Μορφή Kowalski

- ❖ Μια εναλλακτική διατύπωση του κανόνα της ανάλυσης αφορά την μορφή Kowalski

$$\begin{array}{c} p_1, p_2, \dots, p_m \rightarrow q \\ z_1, z_2, \dots, z_n, q' \rightarrow s \\ \hline \theta(p_1, p_2, \dots, p_m, z_1, z_2, \dots, z_n \rightarrow s) \end{array}$$

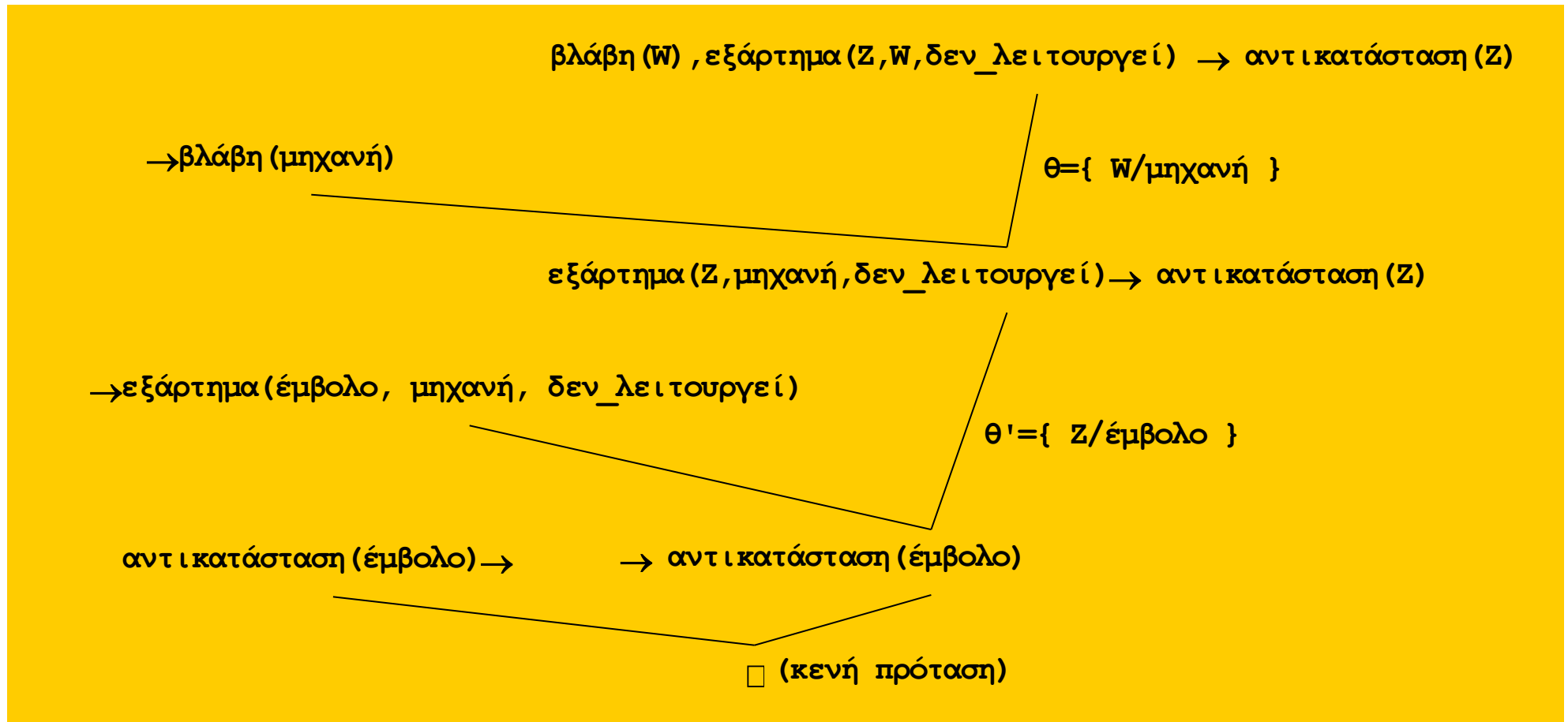
- ❖ Η διαδικασία είναι περισσότερο κατανοητή.
- ❖ Έτσι το προηγούμενο παράδειγμα σε μορφή Kowalski γράφεται:
 - βλάβη (μηχανή)
 - εξάρτημα (έμβολο, μηχανή, δεν_λειτουργεί)
 - σύμπτωμα (έμβολο, θόρυβος)
 - μέρος (έμβολο, μηχανή)
 - βλάβη (W) , εξάρτημα (Z, W, δεν_λειτουργεί) → αντικατάσταση (Z)
 - βλάβη (X) , σύμπτωμα (R, θόρυβος) , μέρος (R, X) → αντικατάσταση (R)
- ❖ Η άρνησης της προς απόδειξη πρότασης αναπαριστάται στην μορφή Kowalski ως αντικατάσταση (έμβολο) →

Απόδειξη με προτάσεις στην μορφή Kowalski



Παρατηρήσεις στην Διαδικασία Απόδειξης

❖ Αποδείξεις δεν είναι μοναδικές.



Απόδειξη και Αναζήτηση

- ❖ Η εύρεση απόδειξης αποτελεί ουσιαστικά ένα πρόβλημα αναζήτησης
 - ❑ Ο μοναδικός τελεστής μετάβασης είναι ο κανόνας της ανάλυσης
- ❖ Αναζήτηση κατά πλάτος (breadth-first),
 - ❑ Παράγει τις συντομότερες αποδείξεις.
 - ❑ Δραματική αύξηση αριθμού των κόμβων.
- ❖ Εμφανίζεται πρόβλημα της συνδυαστικής έκρηξης.
- ❖ Γραμμική ανάλυση (*linear resolution*),
 - ❑ Τουλάχιστον μία από τις προτάσεις σε κάθε απόπειρα ανάλυσης πρέπει να είναι είτε
 - πρόταση εισόδου ή
 - προηγούμενο αναλυθέν στο συγκεκριμένο κλαδί του δένδρου αναζήτησης.

Πλεονεκτήματα και Μειονεκτήματα της "αρχής της Ανάλυσης"

- ❖ Βάση για την δημιουργία του λογικού προγραμματισμού.
 - ❑ Γλώσσα Prolog
 - ❑ Χρησιμοποιεί προτάσεις Horn και SLD - ανάλυση.

Πλεονεκτήματα και Μειονεκτήματα της Κατηγορηματικής Λογικής

- ❖ Πλεονεκτήματα της κατηγορηματικής λογικής:
 - ❑ Αντιστοιχία με τη φυσική γλώσσα, ικανοποιητική έκφραση ποσοτικοποίησης των εννοιών με τους κατάλληλους ποσοδείκτες, ικανότητα να συλλάβει τη γενικότητα.
- ❖ Μειονεκτήματα
 - ❑ *Αδυναμία έκφρασης ασάφειας*: Κάθε πρόταση μπορεί να είναι μόνο αληθής ή ψευδής.
 - ❑ *Αθροιστικότητα των αποτελεσμάτων*: Ένα συμπέρασμα προστίθεται στη γνώση χωρίς να δίνεται η δυνατότητα αναθεώρησής του αν αργότερα κριθεί ότι είναι εσφαλμένο.
- ❖ Λογικός Προγραμματισμός
- ❖ Η κλασική λογική επεκτείνεται με διάφορες εντολές ελέγχου και λειτουργικές δομές,
- ❖ Ευελιξία και ευκολία στην ανάπτυξη εφαρμογών.
- ❖ Χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι η γλώσσα Prolog.

Μη-μονότονη λογική (1/2)

- ❖ Σε μια *μονότονη* λογική, υπάρχει ένα σύστημα αξιωμάτων S (η αρχική βάση γνώσης) και ένα σύνολο τύπων F που αποδεικνύονται (συνάγονται) από το S .
- ❖ Η προσθήκη ενός ή περισσότερων αξιωμάτων στο S (απόκτηση νέας γνώσης), το σύνολο F αυξάνει *μονότονα*.
- ❖ Πλεονεκτήματα:
 - ❑ Κάθε φορά που προστίθεται ένα νέο γεγονός στο S , δε χρειάζονται νέοι έλεγχοι.
 - ❑ Για κάθε νέο γεγονός που αποδεικνύεται δεν είναι απαραίτητη η καταγραφή των γεγονότων πάνω στα οποία βασίζεται η αλήθεια του.
- ❖ Μειονεκτήματα:
 - ❑ η προσθήκη νέων αξιωμάτων μπορεί να μειώσει το σύνολο των δυνατών συμπερασμάτων, αφαιρώντας κάποια που αποδεικνύονται εσφαλμένα μετά την προσθήκη
- ❖ Οι μη-μονότονες συλλογιστικές είναι κατάλληλες για την αντιμετώπιση κάποιων καταστάσεων που εμφανίζονται συχνά στον πραγματικό κόσμο:
 - ❑ Καταστάσεις για τις οποίες δεν έχουμε πλήρη γνώση
 - ❑ Καταστάσεις στις οποίες, η γνώση δημιουργείται κατά τη διάρκεια της εκτέλεσης ενεργειών, για τις οποίες δεν είμαστε βέβαιοι για την αναγκαιότητα ή ορθότητά τους.
 - ❑ Καταστάσεις στις οποίες η γνώση μεταβάλλεται.

- Καταστάσεις στις οποίες το σύστημα χρησιμοποιεί υποθέσεις (assumptions).

Μη-μονότονη λογική (2/2)

- ❖ Στη μη-μονότονη τροπική λογική (*non-monotonic modal logic*) εισάγεται ένας νέος τροπικός τελεστής ο οποίος δηλώνει ότι ένα γεγονός "είναι συνεπές με τις τρέχουσες πεποιθήσεις".
- ❖ Η συλλογιστική εύλογων υποθέσεων (*default reasoning*) χρησιμοποιείται σε περιπτώσεις κατά τις οποίες ένα γεγονός συνάγεται από ένα δοσμένο γεγονός, γιατί έτσι συμβαίνει συνήθως και γιατί δεν υπάρχει ένδειξη για το αντίθετο.
- ❖ Το πρόβλημα της μονοτονίας αντιμετωπίζεται με την εισαγωγή κατάλληλων μηχανισμών εξαγωγής συμπερασμάτων οι οποίοι καταγράφουν ποια γεγονότα χρησιμοποιήθηκαν για την εξαγωγή ενός νέου συμπεράσματος.
- ❖ Τα συστήματα που χρησιμοποιούν αυτούς τους μηχανισμούς ονομάζονται *συστήματα συντήρησης αλήθειας (truth maintenance systems)*.
- ❖ TMS (McAllester 1980): Διατηρεί συνεχώς τη συνέπεια ενός συνόλου λογικών ισχυρισμών, ώστε να βρεθεί κάποια λύση σε ένα πρόβλημα ικανοποίησης περιορισμών.
- ❖ ATMS (De Kleer 1986): Δίνει τη δυνατότητα εύρεσης περισσότερων εναλλακτικών λύσεων μέσω της συλλογιστικής σε παράλληλους κόσμους
- Οι κόσμοι είναι εσωτερικά συνεπείς, αλλά μεταξύ τους μπορεί να είναι ασυνεπείς.

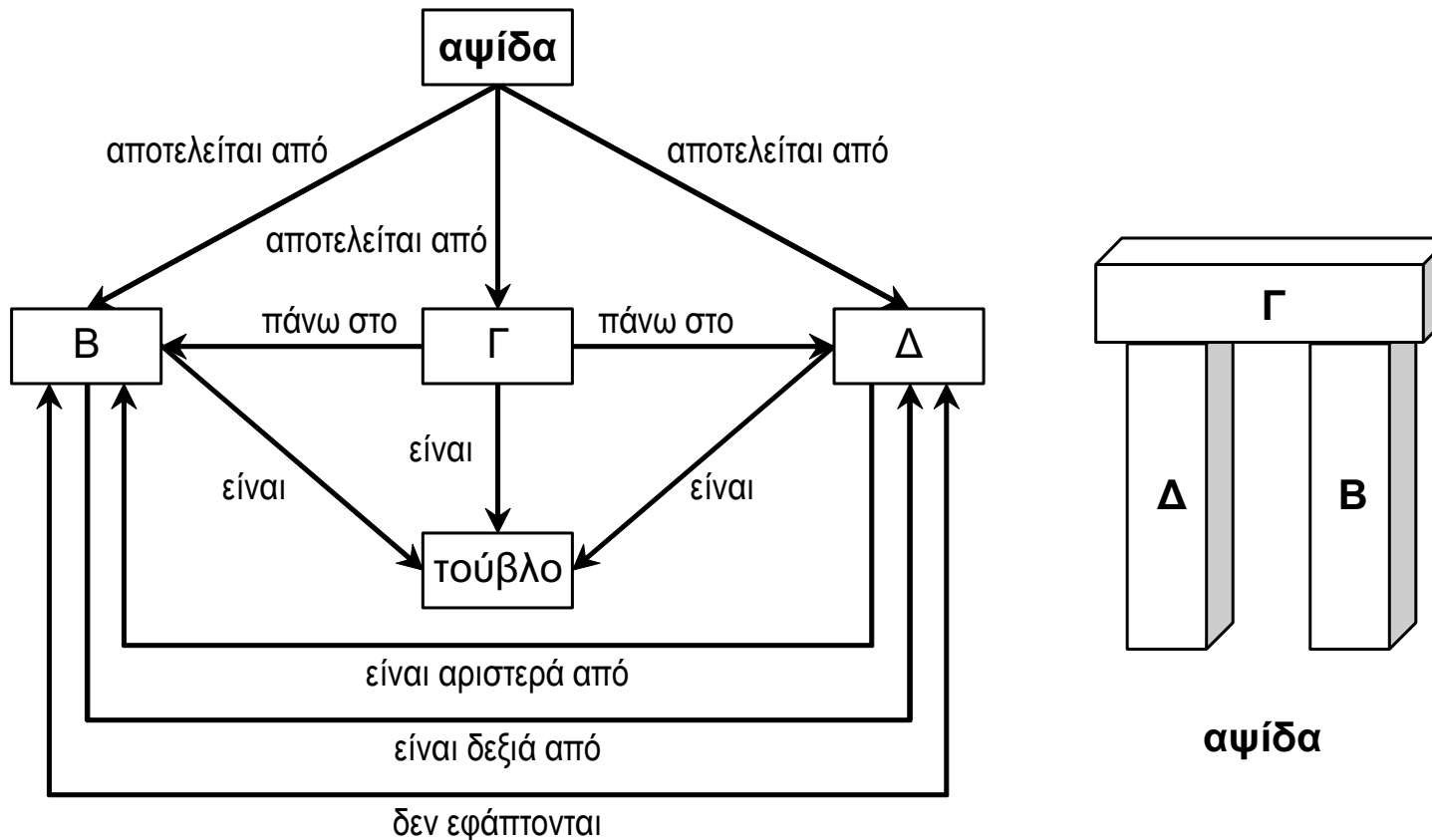
Δομημένες Αναπαραστάσεις Γνώσης

- ❖ Κλασική Λογική: αυστηρότητα στην αναπαράσταση της γνώσης
- ❖ Στην πράξη:
 - ❑ απαιτείται μια λιγότερο αυστηρή και περισσότερο διαισθητική προσέγγιση
 - ❑ είναι επιθυμητή η μείωση του όγκου που καταλαμβάνει η γνώση

- ❖ Δομημένες Αναπαραστάσεις Γνώσης
 - ❑ Σημασιολογικά Δίκτυα
 - ❑ Πλαίσια
 - ❑ Αντικείμενα
 - ❑ Εννοιολογική Εξάρτηση
 - ❑ Σενάρια
 - ❑ Εννοιολογικοί Γράφοι

Σημασιολογικά Δίκτυα (semantic networks)

- ❖ Αποτελούνται από κόμβους (nodes) και δεσμούς (links) ανάμεσά τους.
- ❑ **κόμβοι:** υποδηλώνουν κλάσεις αντικειμένων (classes), αντικείμενα (objects), έννοιες (concepts), τιμές ιδιοτήτων (values)
- ❑ **δεσμοί:** υποδηλώνουν σχέσεις (relations) μεταξύ αντικειμένων ή ιδιότητες που συνδέουν αντικείμενα με τιμές.

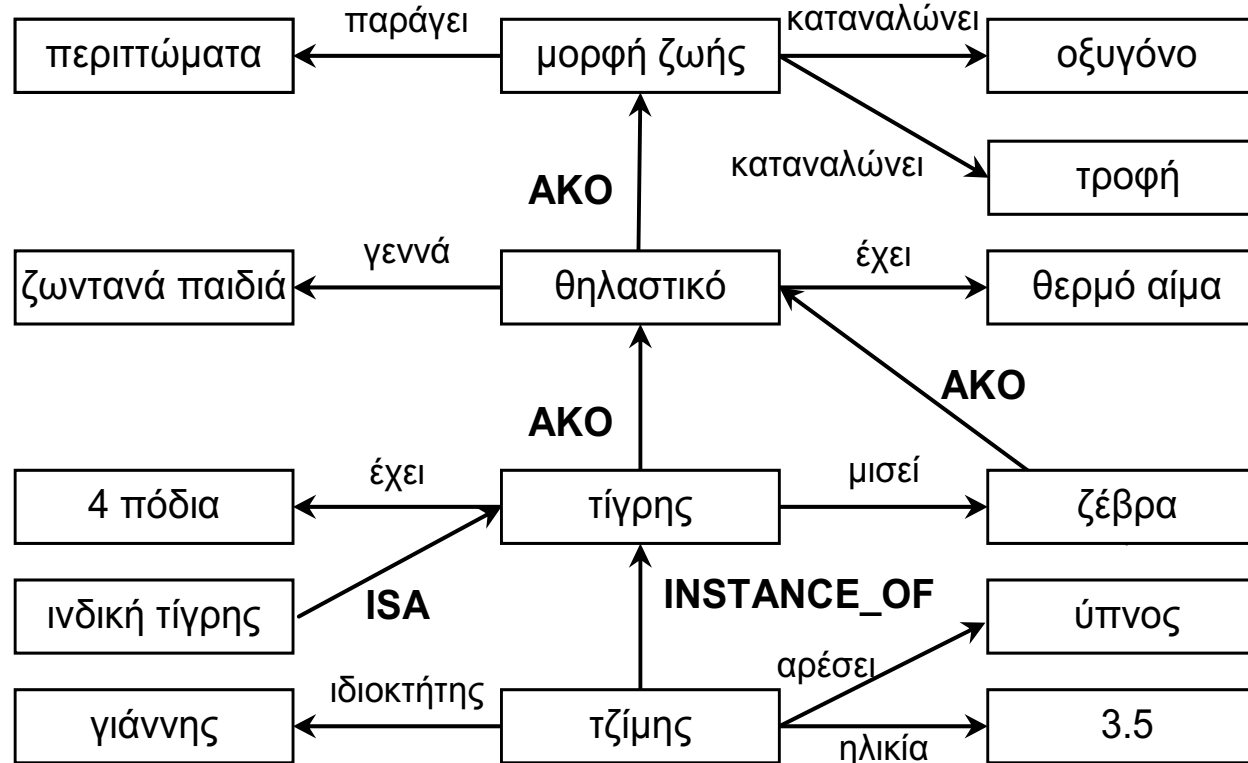


Η Ιεραρχική Δομή των Σημασιολογικών Δικτύων

- ❖ Σημαντικότερα είδη δεσμών: AKO, ISA, INSTANCE_OF.
- ❖ Η σχέση **AKO** υπάρχει μεταξύ κλάσεων αντικειμένων. Σε κόμβο που συνδέεται με σχέση AKO με κάποιον άλλον μπορούν να προστεθούν νέοι δεσμοί που προσδίδουν νέες ιδιότητες.
 - ❑ η κλάση "τίγρης" είναι AKO της κλάσης "θηλαστικό"
- ❖ Η σχέση **ISA** είναι παρόμοια με τη σχέση AKO, με τη διαφορά ότι δε μπορεί να προστεθούν νέες ιδιότητες παρά μόνον να κληρονομηθούν οι ήδη υπάρχουσες ιδιότητες από κόμβους ψηλότερα στην ιεραρχία ή οι ιδιότητες αυτές να αλλάξουν τιμές.
 - ❑ η κλάση "ινδική τίγρης" είναι ISA της κλάσης "τίγρης"
- ❖ Η σχέση **INSTANCE_OF** υπάρχει μόνο μεταξύ κόμβων αντικειμένων και κόμβων γενικότερων κλάσεων.
 - ❑ δεν μπορεί να αποτελεί υπερκλάση άλλης κλάσης.
 - ❑ ο συγκεκριμένος τίγρης "τζίμης" είναι INSTANCE_OF της κλάσης "τίγρης"

Κληρονομικότητα στα Σημασιολογικά Δίκτυα

- ❖ Ένα αντικείμενο κληρονομεί ιδιότητες από μία υψηλότερη ιεραρχικά κλάση.



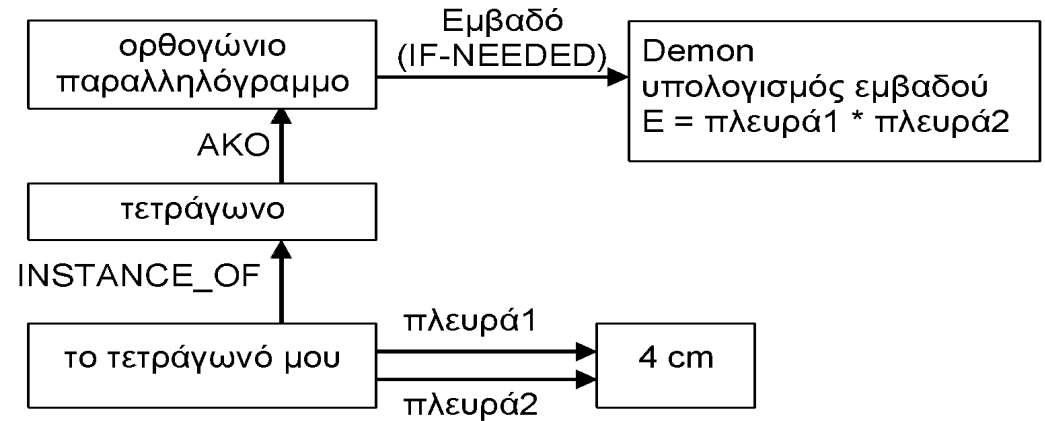
- ❖ Για το συγκεκριμένο τίγρη ("τζίμη") δε χρειάζεται να δηλωθούν παρά μόνον τα χαρακτηριστικά αυτά που είναι αποκλειστικά δικά του ("ιδιοκτήτης", "ηλικία", "αρέσει").

- ❖ Τα υπόλοιπα χαρακτηριστικά-ιδιότητες κληρονομούνται

Προσκόλληση Διαδικασιών

❖ Καλούνται για να δώσουν κάποιο αποτέλεσμα, *μόνον εάν χρειάζεται (IF-NEEDED)*.

□ ονομάζονται και *δαίμονες (daemons)*.



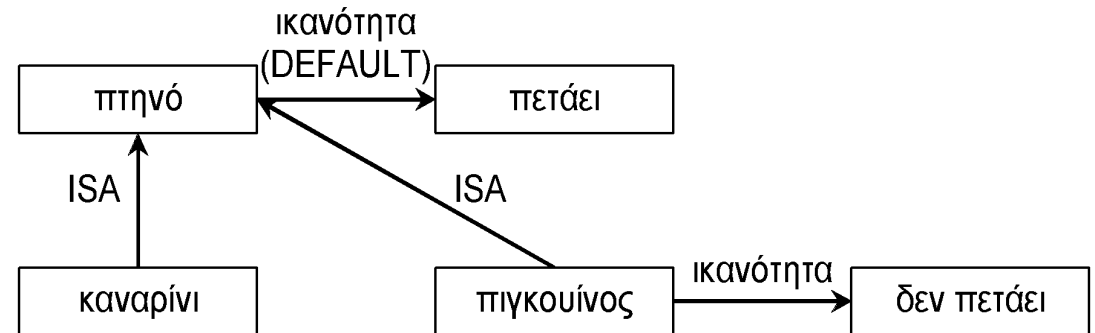
Προκαθορισμένες Τιμές και Εξαιρέσεις

❖ Η συνήθης τιμή μιας ιδιότητας ονομάζεται *προκαθορισμένη τιμή (DEFAULT)*

□ Χρησιμοποιείται όταν δεν υπάρχουν πληροφορίες για την τιμή μιας ιδιότητας.

□ Μπορεί, αν χρειαστεί, να αλλάξουν σε άλλες κλάσεις ή αντικείμενα, χαμηλότερα στην ιεραρχία

□ Τρόπος υλοποίησης συλλογιστικής των εύλογων υποθέσεων.



Πλεονεκτήματα και Μειονεκτήματα

- ❖ Συμπαγής αναπαράσταση (κληρονομικότητα, προσκολλημένες διαδικασίες)
- ❖ Δυνατότητα εξαγωγής συμπερασμάτων (λόγω της ιεραρχικής δομής).

- ❖ Η γνώση που περιγράφεται είναι πολλές φορές διάσπαρτη μέσα στα δίκτυα:
 - ❑ η αναζήτηση παίρνει πολύ χρόνο
 - ❑ η παραμικρή αλλαγή μπορεί να επιφέρει σημασιολογικές αλλαγές

- ❖ **Ευριστική ανεπάρκεια** (heuristic inadequacy)
 - ❑ κάθε φορά που γίνεται αναζήτηση συγκεκριμένης πληροφορίας μπορεί να εμφανιστεί το φαινόμενο της *συνδυαστικής έκρηξης* (*combinatorial explosion*).
- ❖ **Λογική ανεπάρκεια** (logical inadequacy)
 - ❑ η σημασιολογία των αντικειμένων δεν είναι ξεκάθαρη
 - ❑ ένα "αυτοκίνητο" στο σημασιολογικό δίκτυο μπορεί να εκληφθεί ως οποιοδήποτε αυτοκίνητο (κλάση) ή ένα συγκεκριμένο αυτοκίνητο (αντικείμενο)

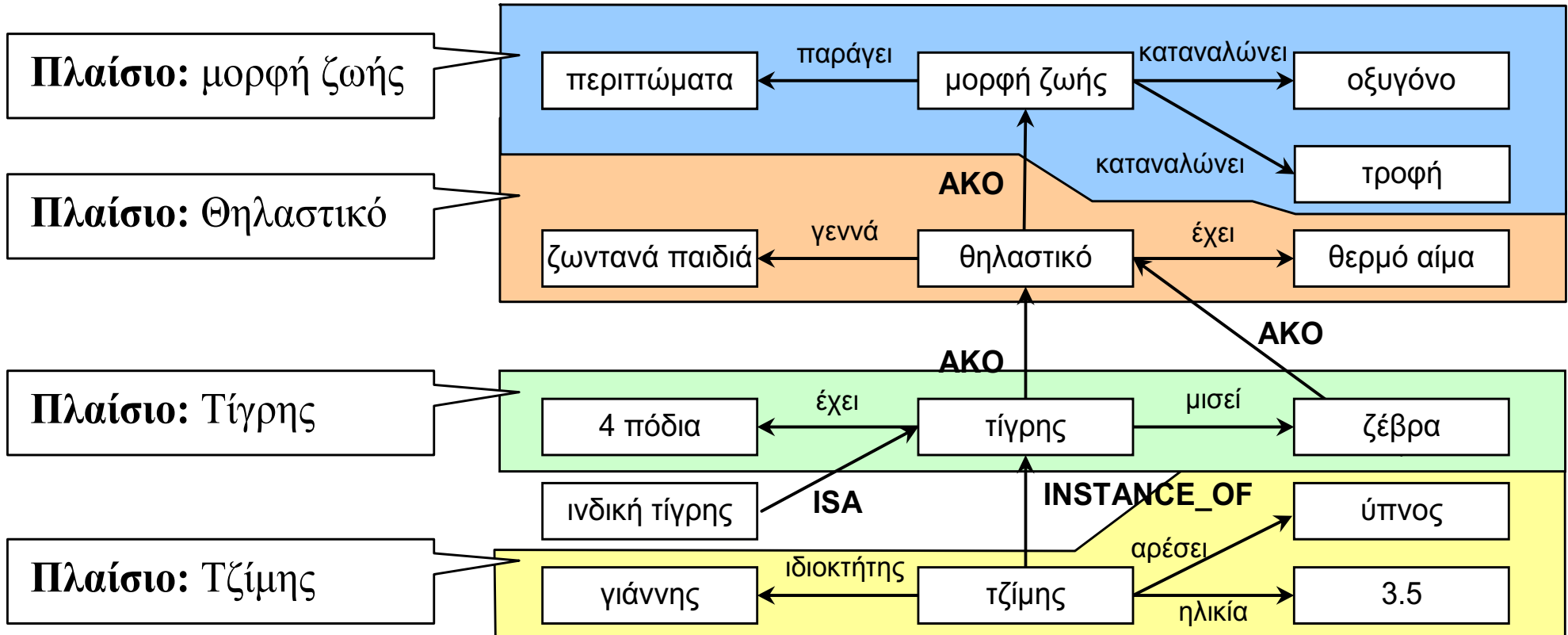
- ❖ **Βασικότερο μειονέκτημα:** δεν υπάρχουν προκαθορισμένες δομικά σχέσεις πάνω στις οποίες μπορεί να στηριχτεί ο σχεδιασμός τους.

Πλαίσια (frames)

- ❖ Ορίστηκαν από τον Minsky σαν "**δομές δεδομένων για την αναπαράσταση στερεότυπων καταστάσεων**". Ονομάζονται και *σχήματα (schemata)*.
- ❖ Τα πλαίσια έχουν:
 - ❑ όνομα,
 - ❑ μία σειρά από **ιδιότητες (slots)** που συνδέονται άμεσα με τις **τιμές τους (fillers)**,
 - προκαθορισμένες τιμές
 - όψεις (*facets*)
 - ❑ **προσκολλημένες διαδικασίες** (όχι υποχρεωτικά) που ονομάζονται **δαίμονες (demons)**
 - μπορεί να ενεργοποιούνται όταν τα πλαίσια μεταβάλλονται για κάποιο λόγο
- ❖ Αν και απαιτούν επιδεξιότητα και επίπονη εργασία, εξελίχθηκαν σε έναν σημαντικό τρόπο αναπαράστασης γνώσης.

Ιεραρχία στα Πλαίσια

- ❖ Τα πλαίσια μπορούν να αποτελέσουν αντικείμενα-κόμβους ενός σημασιολογικού δικτύου και να συνδεθούν με μία ιεραρχία.



Πλαίσια και Σημασιολογικά Δίκτυα

- ❖ Τα πλαίσια (όπως και τα σημασιολογικά δίκτυα) παρέχουν:
 - ❑ κληρονομικότητα,
 - ❑ προσκόλληση διαδικασιών και
 - ❑ προκαθορισμένες τιμές

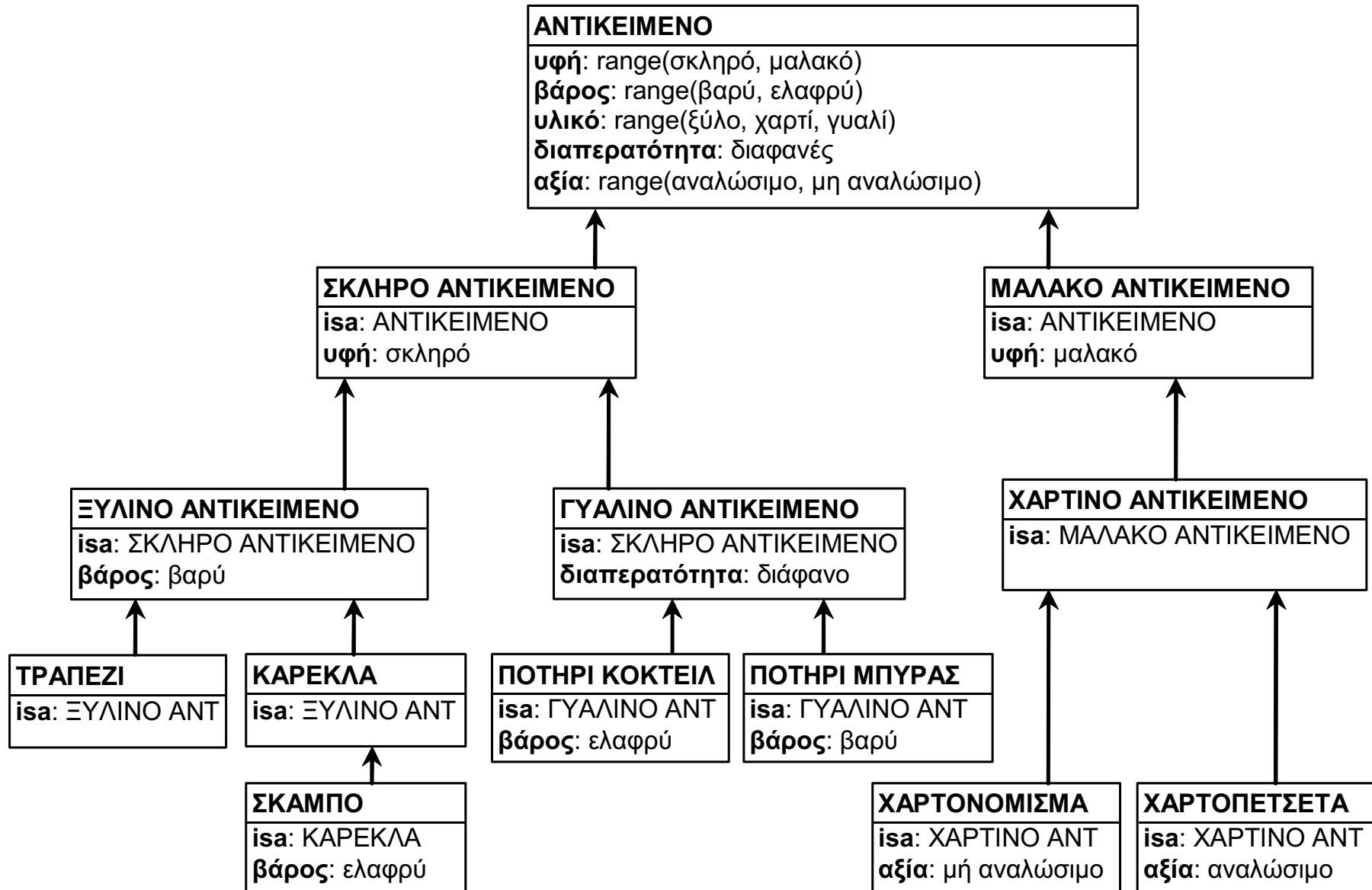
- ❖ Τα πλαίσια υπερτερούν έναντι των σημασιολογικών δικτύων στο ότι:
 - ❑ Η ιεραρχία των εννοιών στα πλαίσια είναι πιο ξεκάθαρη:
 - Ένα πλαίσιο περιέχει **όλη** την πληροφορία για τη συγκεκριμένη έννοια που αναπαριστά.
 - Ένας κόμβος σε ένα σημασιολογικό δίκτυο αναπαριστά **μόνο** την έννοια, ενώ οι ιδιότητές της περιγράφονται σε **άλλους** κόμβους που συνδέονται με αυτόν.
 - ❑ παράκαμψη *λογική ανεπάρκειας*
 - δεν εμφανίζουν τόσο έντονο το φαινόμενο της συνδυαστικής έκρηξης στην αναζήτηση

Πλαίσια και Εγγραφές

- ❖ Τα πλαίσια θυμίζουν ίσως τις εγγραφές (*records*) των γλωσσών προγραμματισμού.
- ❖ Όμως τα πλαίσια:
 - ❑ δεν είναι κατ' ανάγκη όμοια μεταξύ τους, όπως οι εγγραφές,
 - ❑ δεν περιέχουν ίδιου τύπου πληροφορίες, ούτε μόνον απλά δεδομένα,
 - ❑ οργανώνονται σε ιεραρχικές δομές υποστηρίζοντας ταυτόχρονα κληρονομικότητα, και
 - ❑ μπορεί να έχουν προσκολλημένες διαδικασίες.

- ❖ μοιάζουν περισσότερο με τα αντικείμενα (*objects*) του αντικειμενοστραφούς προγραμματισμού και λιγότερο με τις εγγραφές

Παράδειγμα Ιεραρχίας Πλαισίων



Εξαγωγή Συμπερασμάτων με Πλαίσια

- ❖ Μπορεί να υλοποιηθεί με οποιονδήποτε από τους μηχανισμούς συμπερασμάτων της λογικής.
- ❖ Επιπλέον χρειάζεται ένας τρόπος για την ανάκληση της τιμής μίας ιδιότητας:

Διαδικασία `find(Frame, Attribute, Value)`

Αν η ιδιότητα `Attribute` υπάρχει στο πλαίσιο `Frame`,
τότε

επέστρεψε την τιμή της `Value`

Αλλιώς,

ακολούθησε την ιεραρχία δεσμών `ISA` ή `AKO` ή `INSTANCE_OF` και επανέλαβε τη διαδικασία με νέο πλαίσιο `NewFrame` το αμέσως παραπάνω πλαίσιο του `Frame` στην ιεραρχίας, δηλ. `find(NewFrame, Attribute, Value)`.

- ❖ Λόγω δυνατότητας ύπαρξης **πολλαπλής κληρονομικότητας** (*multiple inheritance*), πρέπει να καθορίζει και από ποιο πλαίσιο θα κληρονομηθεί η ιδιότητα.
 - π.χ. η αναζήτηση πρώτα σε πλάτος θα επιστρέψει την τιμή της ιδιότητας που βρίσκεται στο πιο κοντινό πλαίσιο.

Αντικείμενα (objects)

- ❖ Τρόπος αναπαράστασης δομημένης γνώσης που προήλθε από την έρευνα για γλώσσες προσομοίωσης.
 - ❑ ανάγκη δημιουργίας εύκολα αντιληπτών μοντέλων
 - ❑ αποδόμηση (decomposition) σε αντικείμενα σχετικά με οντότητα του φυσικού κόσμου

Βασικά χαρακτηριστικά αντικειμένων:

- ❑ έχουν κάποια κατάσταση (**ιδιοότητες** *properties-fields*)
- ❑ έχουν ένα σύνολο από **μεθόδους** (*methods*) που ορίζουν τη συμπεριφορά του αντικειμένου
- ❑ αντιδρούν σε προκαθορισμένα **μηνύματα** ή **γεγονότα** (*messages* ή *events*) που λαμβάνουν από τον εξωτερικό κόσμο

Οργάνωση Αντικειμένων

❖ κλάση (*class*) αντικειμένου: το σύνολο των ιδιοτήτων των μεθόδων και των μηνυμάτων στα οποία αυτό αποκρίνεται

❖ Χαρακτηριστικά:

- ❑ απόκρυψη εσωτερικής πολυπλοκότητας (*εγκλεισμός – encapsulation*)
- ❑ πρότυπα για την παραγωγή στιγμιότυπων (*instances*)
- ❑ απόκρυψη εσωτερικής πληροφορίας και μάλιστα με διαβαθμισμένο τρόπο
- ❑ οργάνωση σε ιεραρχίες (π.χ. "όχημα" γενικότερη από "αυτοκίνητο")
- ❑ κληρονομικότητα (*inheritance*)
 - αποφυγή της επανάληψης ορισμού κοινών χαρακτηριστικών και μεθόδων
- ❑ πολυμορφισμός (*polymorphism*)
- ❑ πολλαπλή κληρονομικότητα (*multiple inheritance*)

Αντικείμενα και Πλαίσια

- ❖ **Ομοιότητες:** δομημένη περιγραφή, κληρονομικότητα, ιδιότητες-slots, μέθοδοι-δαίμονες.
- ❖ **Βασικές Διαφορές**
 - ❑ Οι κλάσεις στα πλαίσια δεν παίζουν το ρόλο του τύπου δεδομένων, όπως στα αντικείμενα, αλλά το ρόλο του προτύπου.
 - ❑ Τα στιγμιότυπα στα πλαίσια δεν είναι υποχρεωτικό να ακολουθήσουν αυστηρά τις προδιαγραφές που ορίζει η κλάση τους.
 - ❑ Η αυστηρά καθορισμένη δομή των αντικειμένων, διευκολύνει τον έλεγχο ορθότητας των προγραμμάτων και επιτρέπει την παραγωγή αποδοτικότερου κώδικα.
 - ❑ Η πρόσβαση στις ιδιοτήτων και τις μεθόδους των αντικειμένων είναι *διαβαθμισμένη* ενώ στα πλαίσια οι τιμές των *ιδιοτήτων (slots)* είναι πάντα προσβάσιμες.
 - ❑ Τα αντικείμενα εμπεριέχουν τον κώδικα ελέγχου μέσα τους, με τη μορφή μεθόδων. Στα πλαίσια ο υπόλοιπος κώδικας είναι αποθηκευμένος εκτός των πλαισίων.
 - ❑ Στα πλαίσια, οι δαίμονες ενεργοποιούνται αυτόματα όταν γίνει πρόσβαση στις ιδιότητες. Στα αντικείμενα οι μέθοδοι, ενεργοποιούνται με την εκούσια αποστολή μηνυμάτων από τους χρήστες ή από μεθόδους άλλων αντικειμένων.

Εννοιολογική Εξάρτηση (conceptual dependency) (1/2)

- ❖ Προέκυψε (Schank, '74) ως αποτέλεσμα των προσπαθειών να ενσωματωθούν οι βασικές σημασιολογικές σχέσεις της φυσικής γλώσσας στον ίδιο το φορμαλισμό, παρά να αποτελούν τμήμα του πεδίου γνώσης (*domain knowledge*).
- ❖ Υπάρχουν τέσσερις **αρχέγονες εννοιολογικές μορφές** (*primitive conceptualizations*) πάνω στις οποίες μπορεί να βασιστεί η ερμηνεία.
 - ❑ **ACTs** (ενέργειες), **PPs** (αντικείμενα), **AAs** (Προσδιορισμοί Ενεργειών), **PAs** (Προσδιορισμοί Αντικειμένων)
 - ❑ αυτές αναλύονται περαιτέρω ώστε να καλύψουν περισσότερο εξειδικευμένες περιπτώσεις
 - ❑ Π.χ. ATRANS: μεταφορά μιας αφηρημένης σχέσης (π.χ. «δίνω»)
 - ❑ Π.χ. PTRANS: μεταφορά φυσικής θέσης αντικειμένου (π.χ. «πηγαίνω»)
- ❖ Υπάρχουν **σύμβολα** για τον ακριβέστερο καθορισμό χρόνου και τρόπου.
 - ❑ π.χ. **p** (παρελθόν), **f** (μέλλον), **c** (υπό προϋποθέσεις), κτλ.
- ❖ Τα αρχέγονα στοιχεία και σύμβολα χρησιμοποιούνται για να οριστούν σταθερές και με καλά ορισμένη σημασιολογία, σχέσεις μεταξύ των εννοιών (**σχέσεις εννοιολογικής εξάρτησης** - *conceptual dependency relationships*)

Εννοιολογική Εξάρτηση (conceptual dependency) (2/2)

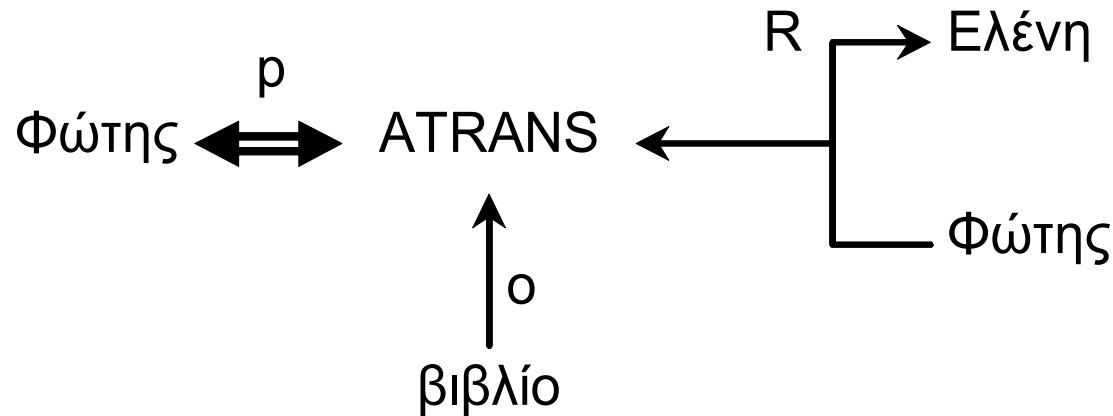
- ❖ Οι σχέσεις εννοιολογικής εξάρτησης είναι *εννοιολογικοί συντακτικοί κανόνες*:
 - ❑ εύκολη εξαγωγή συμπερασμάτων από προτάσεις
 - ❑ η αναπαράσταση δεν εξαρτάται από τη γλώσσα διατύπωσης της πρότασης

Παράδειγμα Σχέσεων Εννοιολογικής Εξάρτησης

Σχέση	Ερμηνεία
$PP \Leftrightarrow ACT$	Κάποιος ενεργεί.
$ACT \xleftarrow{O} PP$	Το αντικείμενο κάποιας ενέργειας.
$ACT \xleftarrow{R} \begin{matrix} PP \\ PP \end{matrix}$	Ο δότης και ο παραλήπτης ενός αντικειμένου σε μία ενέργεια.

Γράφοι Εννοιολογικής Εξάρτησης

❖ **Παράδειγμα:** Έστω η πρόταση: "Ο Φώτης έδωσε το βιβλίο στην Ελένη".



- ❑ τα βέλη δείχνουν την κατεύθυνση της εξάρτησης,
- ❑ το διπλό βέλος " \Leftrightarrow " σημαίνει ότι υπάρχει αμφίδρομη σχέση μεταξύ του δράστη και της πράξης,
- ❑ το "*p*" σημαίνει παρελθόν (past),
- ❑ το "*ΑΤΡΑΝS*" σημαίνει μεταφορά κτήσης ενός πράγματος (abstract transitive),
- ❑ το "*o*" δηλώνει το αντικείμενο (object) και
- ❑ το "*R*" δηλώνει τον παραλήπτη (recipient).

Πλεονεκτήματα - Μειονεκτήματα

- ❖ Αυστηρά ορισμένη μεθοδολογία.
- ❖ Προτάσεις που έχουν το ίδιο νόημα θα αναπαρίστανται με παρόμοιο τρόπο.
 - ❑ ταίριασμα δύο γράφων \implies δύο προτάσεις έχουν το ίδιο νόημα.
- ❖ Πλήρως δοκιμασμένο μοντέλο αλλά όχι για γενικές περιπτώσεις.
- ❖ Συχνά, από μικρές φράσεις παράγονται τεράστιοι γράφοι που απαιτούν ανάλογους υπολογιστικούς πόρους για το χειρισμό τους.
- ❖ Κάποιες κοινές έννοιες της φυσικής γλώσσας είναι δύσκολο να κωδικοποιηθούν:
 - ❑ π.χ. οι έννοιες "λίγο", "χαμηλή", "βαρύς", κτλ.
 - ❑ σημασιολογικά ασαφείς
 - ❑ ο χειρισμός τους γίνεται καλύτερα με άλλες μεθόδους όπως η ασαφής λογική.

Σενάρια (scripts)

- ❖ *στερεότυπες καταστάσεις (stereotypical situations)*
 - ❑ οργάνωση γνώσης στον ανθρώπινο νου.
 - ❑ π.χ. η φράση "πήγα στο εστιατόριο" υπονοεί χωρίς να το δηλώνει ρητά ότι π.χ. "κάθισα σε ένα τραπέζι, παρήγγειλα, έφαγα, πλήρωσα, κτλ"

- ❖ Τα σενάρια είναι μία στερεότυπη ακολουθία γεγονότων που περιγράφουν τυπικές καταστάσεις σε συγκεκριμένα πλαίσια δραστηριότητας.

- ❖ Βασικά μέρη:
 - ❑ *Συνθήκες εισόδου (entry conditions)*
 - ❑ *Αποτελέσματα (results)*
 - ❑ *Σκηνικά (props)*
 - ❑ *Ρόλους (roles)*
 - ❑ *Παραπομπές (track)*
 - ❑ *Σκηνές (scenes)*

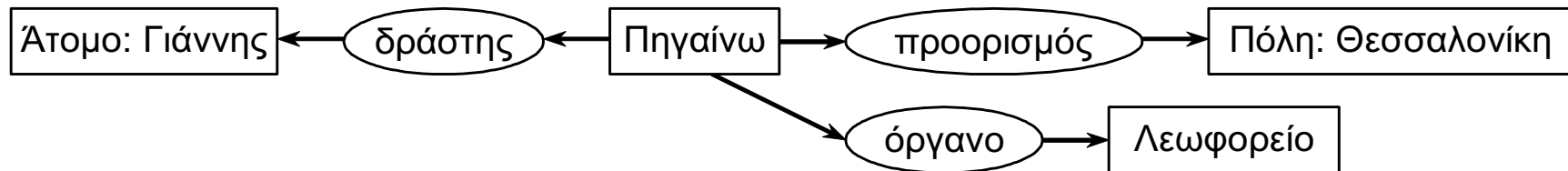
Παράδειγμα Σεναρίου (Εστιατόριο)

<p>Σενάριο: Εστιατόριο Παραπομπή: Καφετερία</p>	<p>ΣΚΗΝΗ 1: Είσοδος Πελάτης PTRANS Πελάτη στο εστιατόριο Πελάτης ATTENDS μάτια στα τραπέζια Πελάτης MBUILD που θα καθίσει Πελάτης PTRANS στο τραπέζι</p>
<p>Ιδιότητες/Αντικείμενα: Τραπέζι Μενού Φαγητό Λογαριασμός Χρήματα</p>	<p>ΣΚΗΝΗ 2: Παραγγελία (Μενού στο Τραπέζι) (Σερβιτόρος φέρνει Μενού) Πελάτης PTRANS Μενού στον Πελάτη Πελάτης MBUILD επιλογή Φαγητού Πελάτης MTRANS σήμα στο Σερβιτόρο Σερβιτόρος PTRANS Σερβιτόρο στο Τραπέζι Πελάτης MTRANS "Θέλω Φαγητό X" στο Σερβιτόρο Σερβιτόρος PTRANS Σερβιτόρο στο Μάγειρα Σερβιτόρος ATRANS X στο Μάγειρα Μάγειρας DO (Σενάριο ετοιμασίας φαγητού X)</p>
<p>Ρόλοι: Πελάτης Σερβιτόρος Μάγειρας Ταμίας Ιδιοκτήτης</p>	<p>ΣΚΗΝΗ 3: Φαγητό Μάγειρας ATRANS Φαγητό στο Σερβιτόρο Σερβιτόρος ATRANS Φαγητό στον Πελάτη Πελάτης INGEST Φαγητό</p>
<p>Συνθήκες Εισόδου: Πελάτης πεινάει Πελάτης έχει χρήματα</p>	<p>(Επιλογή: Σκηνή 2 για επόμενη παραγγελία ή αλλιώς Σκηνή 4)</p>
<p>Αποτελέσματα: Πελάτης έχει λιγότερα χρήματα Ιδιοκτήτης έχει περισσότερα χρήματα Πελάτης δεν πεινάει</p>	<p>ΣΚΗΝΗ 4: Έξοδος Σερβιτόρος MOVE (ετοιμάζει Λογαριασμό) Σερβιτόρος PTRANS Σερβιτόρο στον Πελάτη Σερβιτόρος ATRANS Λογαριασμό στον Πελάτη Πελάτης ATRANS Φιλοδώρημα στον Σερβιτόρο Πελάτης PTRANS Πελάτη στον Ταμία Πελάτης ATRANS Χρήματα στον Ταμία Πελάτης PTRANS Πελάτη έξω από το εστιατόριο</p>

- ❖ αντικείμενα και ρόλοι
- ❖ τέσσερις σκηνές
- ❖ συνθήκες εισόδου και αποτελέσματα σεναρίου
- ❖ η αναφορά στο εστιατόριο από μέρος του ανθρώπου ενεργοποιεί το σενάριο του εστιατορίου και συνεπώς ο υπολογιστής μπορεί να κάνει εύλογες υποθέσεις
- ❖ είναι πιθανό η πραγματικότητα να αποκλίνει και να γίνουν λάθος εκτιμήσεις από τον υπολογιστή

Εννοιολογικοί Γράφοι (conceptual graphs)

- ❖ Προτάθηκαν από τον Sowa ('76) και αποτελούν μία γλώσσα αναπαράστασης γνώσης με ρίζες στη μοντέρνα γλωσσολογία, την ψυχολογία και τη φιλοσοφία
- ❖ Συνοδεύονται από κατάλληλες δομές δεδομένων και τεχνικές για το χειρισμό τους.
- ❖ Ένας εννοιολογικός γράφος είναι ένας πεπερασμένος γράφος που αποτελείται από διασυνδεδεμένους **κόμβους εννοιών και σχέσεων**, που **εναλλάσσονται**.



- ❖ Τα τόξα ανήκουν στις σχέσεις και είναι προσκολλημένα στις έννοιες.

Έννοιες

❖ Οι **έννοιες** (*concepts*) αποτελούν στιγμιότυπα των **τύπων-εννοιών** (*concept types*) και αποτελούνται από έναν **τύπο-έννοιας** (πρακτικά μία ετικέτα) συνοδευόμενο συνήθως από μία **αναφορά** (*referent*) σε συγκεκριμένη οντότητα. π.χ. [Άτομο: Γιάννης]

Τύποι Εννοιών

❖ Ένας **τύπος-έννοιας** μπορεί να θεωρηθεί ότι είναι μία κλάση ή κατηγορία οντοτήτων, ενεργειών, ιδιοτήτων, αφηρημένων εννοιών, κτλ.

❖ Μπορεί να οργανωθούν σε μία **ιεραρχία** (*concept type hierarchy*)

❑ Οντότητα \geq Ενέργεια, Άτομο, Δένδρο

❑ Άτομο \geq Φοιτητής, Υπάλληλος

❖ Η ιεραρχία:

❑ υποδηλώνει την ύπαρξη **κληρονομικότητας** στις ιδιότητές τους

❑ ορίζει σχέσεις γενίκευσης και εξειδίκευσης

• π.χ. η έννοια [Φοιτητής] είναι εξειδίκευση της έννοιας [Άτομο]

• ισοδύναμα: η έννοια [Άτομο] είναι γενίκευση της έννοιας [Φοιτητής]

Αναφορά

- ❖ Μια έννοια χωρίς αναφορά (π.χ. [Βιβλίο] ή [Βιβλίο:*]) υποδηλώνει μία υπαρκτή αλλά απροσδιόριστη οντότητα (υπονοείται ο υπαρξιακός ποσοδείκτης "∃").
- ❖ Η αναφορά μπορεί να πάρει πολλές μορφές. Ενδεικτικά:
 - ❑ [Βιβλίο:∇] – κάθε βιβλίο
 - ❑ [Βιβλίο:{"AI","I Robot"}] – τα βιβλία "AI" και "I Robot"
 - ❑ [Βιβλίο:{*}@3] – τρία βιβλία
 - ❑ [Ακέραιος:5] – ο αριθμός 5
 - ❑ [Πόλη:Αθήνα]
 - ❑ [Ηλικία:@<35,έτη>] – παρατίθεται η τιμή και η μονάδα μέτρησης
 - ❑ [Δένδρο:#9143] – συγκεκριμένο δένδρο που είναι καταγεγραμμένο με τον κωδικό 9143 σε κάποιο κατάλογο οντοτήτων
- ❖ Έννοιες συμφραζομένων (*contextual concepts*): ειδικές περιπτώσεις εννοιών
 - ❑ [**Πρόταση**: [Άνθρωπος:∇]→(χαρακτηριστικό)→[Θνητός]]
Υπάρχει μια πρόταση, η 'Κάθε άνθρωπος είναι θνητός'
 - ❑ [**Κατάσταση**: [Κρατούμενος:#17]→(μέσα)→[Κελί:#33]]
Υπάρχει μία κατάσταση: "ο κρατούμενος #17 βρίσκεται στο κελί #33".

Συναναφορά (coreference)

- ❖ δύο ή περισσότερες έννοιες αναφέρονται στην ίδια οντότητα

[Τρώω] →

(δράστης) → [Πίθηκος]

(αντικείμενο) → [Καρύδι : *x]

(όργανο) → [Κουτάλι] → (υλικό) → [Κέλυφος] ← (τμήμα) ← [Καρύδι : ?x]

- ❖ Αναπαράσταση με μία μεταβλητή:

- πρόθεμα "*" για να οριστεί η μεταβλητή (*defining referent*)

- πρόθεμα "?" για να γίνει αναφορά σε συγκεκριμένη μεταβλητή (*bound referent*)

- ❖ Οι έννοιες της συναναφοράς πρέπει να είναι **συμβατές**:

- ίδιου τύπου, ή

- να έχουν κάποια κοινή γενίκευση

Ορισμός νέων τύπων-εννοιών

- ❖ Οι τύποι-εννοιών μπορεί να είναι:
 - ❑ **αρχέγονοι** (*primitive*): αξιωματικά ορισμένοι π.χ. [Τίγρης]
 - ❑ **ορισμένοι** (*defined*): μέσω μονοπαραμετρικών εκφράσεων-λ (*lambda expressions*)

```
type Τίγρης_Τσίρκου(*λ) is
  [Τίγρης:?λ]←(δράστης)←[Εμφανίζεται]→(τοποθεσία)→[Τσίρκο]
"Ο Τίγρης_Τσίρκου είναι ένας τίγρης που εμφανίζεται σε τσίρκο"
```

- ❖ Η ερμηνεία επηρεάζεται από το ποια έννοια αποτελεί την παράμετρο της έκφρασης-λ

```
type Τσίρκο_με_Τίγρη(*λ) is
  [Τίγρης]←(δράστης)←[Εμφανίζεται]→(τοποθεσία)→[Τσίρκο:?λ]
"Τσίρκο_με_Τίγρη είναι ένα τσίρκο στο οποίο εμφανίζεται ένας τίγρης"
```

Εννοιολογικές Σχέσεις (*conceptual relations*)

- ❖ συσχετίζουν τις έννοιες μεταξύ τους
- ❖ μπορεί να οργανωθούν σε ιεραρχία
- ❖ ορίζονται από έναν *τύπο-σχέσης* (*relation type*) που περιλαμβάνει:
 - ❑ ετικέτα (label)
 - ❑ σθένος (valence) - αριθμός εννοιών που συσχετίζονται
 - ❑ υπογραφή (signature) – τύποι εννοιών που συνδέει η σχέση
 - για μια n-αδική σχέση η υπογραφή συμβολίζεται με $\langle t_1, t_2, \dots, t_n \rangle$.
- ❖ Παράδειγμα υπογραφής σχέσης:
 - ❑ ο τύπος-σχέσης **δράστης** έχει **σθένος 2** και υπογραφή **<Ενέργεια, Έμψυχο>**



Ορισμός νέων Εννοιολογικών Σχέσεων

- ❖ Οι εννοιολογικές σχέσεις (όπως και οι τύποι-εννοιών) μπορεί να είναι:
 - ❑ αρχέγονες (*primitive*): αξιωματικά ορισμένες
 - ❑ ορισμένες (*defined*): μέσω εκφράσεων-λ (*lambda expressions*)

- ❖ **Παράδειγμα ορισμού της σχέσης "πηγαίνω_σε":**

```
relation πηγαίνω_σε (*λ1, *λ2) is
  [Άτομο: ?λ1] ← (δράστης) ← [Πηγαίνω] → (προορισμός) → [Πόλη: ?λ2]
```

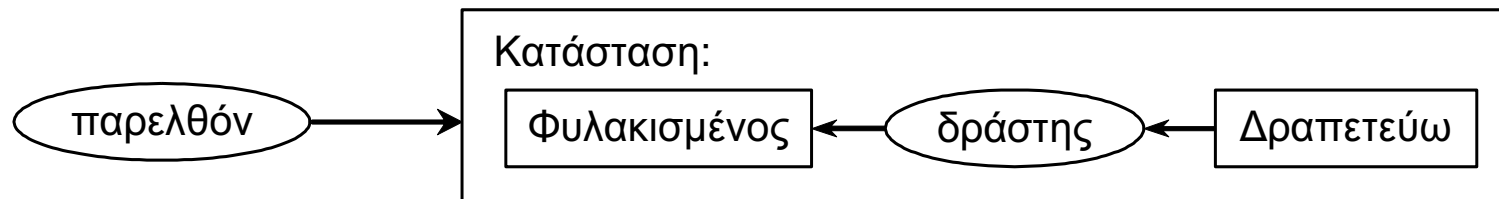
- ❑ **Παράδειγμα χρήσης:** [Άτομο: Φώτης] → (πηγαίνω_σε) → [Πόλη: Φάρσαλα]

- ❖ Μία ορισμένη σχέση (*defined relation*) μέσα σε έναν σημασιολογικό γράφο μπορεί να αντικατασταθεί με τον ορισμό της (έκφραση-λ) και αντίστροφα.

```
[Άτομο: Ελένη] → (πηγαίνω_σε) → [Πόλη: Φάρσαλα]
[Άτομο: Ελένη] ← (δράστης) ← [Πηγαίνω] → (προορισμός) → [Πόλη: Φάρσαλα]
```

Τόξα (*arcs*)

- ❖ συνδέουν τις σχέσεις με τις έννοιες
- ❖ Η φορά των τόξων καθορίζεται από την υπογραφή της σχέσης στην οποία ανήκουν.
 - ❑ τα πρώτα $n-1$ τόξα βάσει της υπογραφής έχουν κατεύθυνση προς τη σχέση, ενώ το τελευταίο απομακρύνεται από τη σχέση
 - ❑ ισχύει και για τις μοναδιαίες σχέσεις
 - "Ένας φυλακισμένος δραπέτευσε"



- ❖ Αρίθμηση Τόξων
 - ❑ με τους αριθμούς 1, 2, κτλ., για τα πρώτα $n-1$ τόξα ($n > 2$) βάσει της υπογραφής
 - ❑ το n -οστό δε χρειάζεται να αριθμηθεί

Εννοιολογικοί Γράφοι και Λογική

- ❖ Η **άρνηση** (NOT): (άρνηση) \rightarrow [Κατάσταση: u]
- Πιο απλά: $\leftarrow [u]$
- **Παράδειγμα:** $\leftarrow [[Βάρκα:\#17] \rightarrow (\text{μέσα}) \rightarrow [\text{Λιμάνι}]]$
- ❖ Η **σύζευξη** (AND): απλή παράθεσή στο ίδιο πλαίσιο συμφραζομένων:

[Πρόταση:

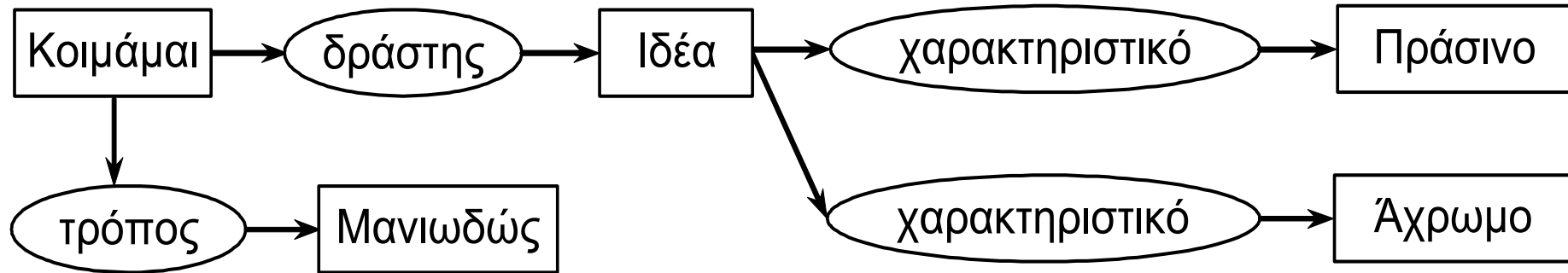
[Άτομο: * x] \rightarrow (ιδιότητα) \rightarrow [Οπλισμένο]

[Άτομο: ? x] \rightarrow (ιδιότητα) \rightarrow [Επικίνδυνο]]

- **Πιο απλά:** η σύζευξη δύο γράφων u και v μπορεί να γραφεί και ως: $[u \ v]$
- ❖ Η **διάζευξη** (OR): βάσει του ότι $p \vee q$ είναι ισοδύναμο με $\leftarrow (\leftarrow p \wedge \leftarrow q)$.
- $\leftarrow [\leftarrow [u] \leftarrow [v]]$
- ❖ **Συνεπαγωγή:** το $p \rightarrow q$ είναι ισοδύναμο με $\leftarrow (p \wedge \leftarrow q)$.
- $\leftarrow [u \leftarrow [v]]$
- Σε μορφή κανόνα: if p and q then w $\leftarrow [p \ q \leftarrow [w]]$ ή $[If: p \ q \ [Then: w]]$

Πώς συνδυάζονται όλα τα προηγούμενα;

❖ Δεν μπορεί όλοι οι αυθαίρετοι συνδυασμοί εννοιών και σχέσεων να παράγουν ορθά δομημένους (well-formed) εννοιολογικούς γράφους. Παράδειγμα:



"Άχρωμες πράσινες ιδέες κοιμούνται μανιωδώς" (Chomsky)

❑ παραβίαση ιεραρχίας και κανόνων κοινής λογικής

❖ Απαιτούνται μηχανισμοί που θα:

❑ εξασφαλίζουν τη δημιουργία ορθά δομημένων εννοιολογικών γράφων

- Κανόνες Ορθής Διαμόρφωσης

❑ επιτρέπουν τη συλλογιστική με αυτούς

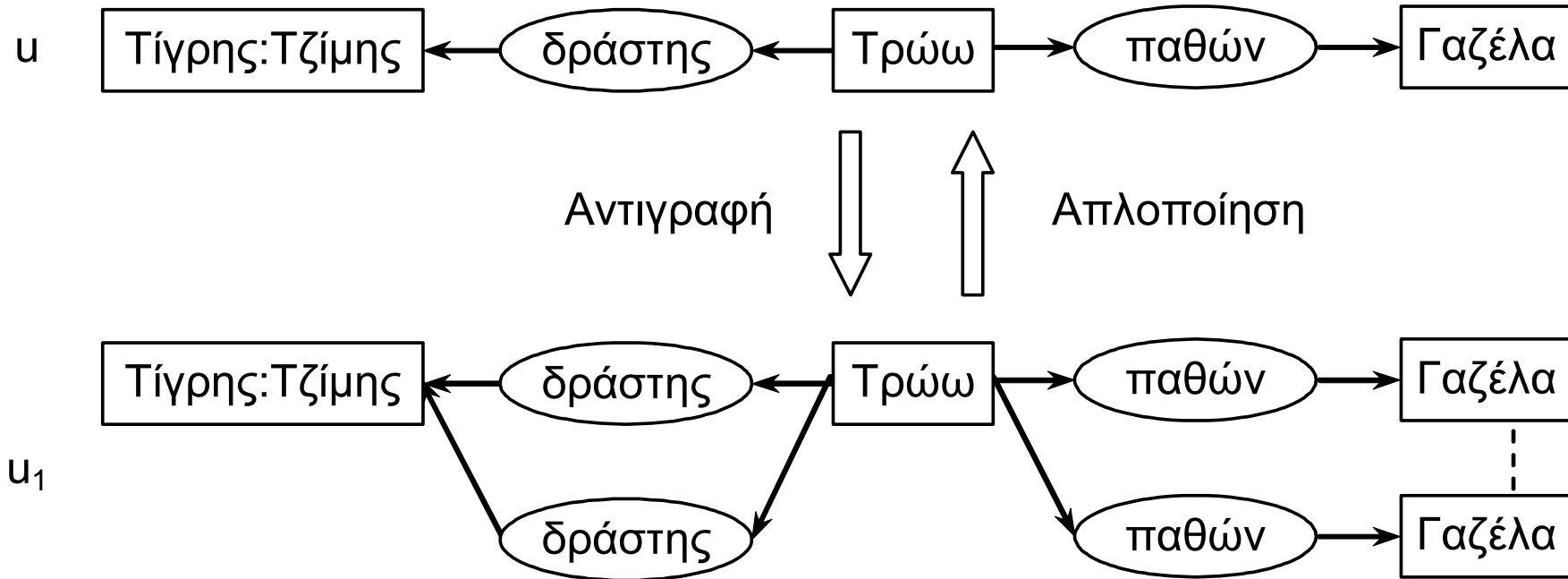
- Κανόνες Εξαγωγής Συμπερασμάτων

Κανόνες Ορθής Διαμόρφωσης (1/2)

(*canonical formation rules*)

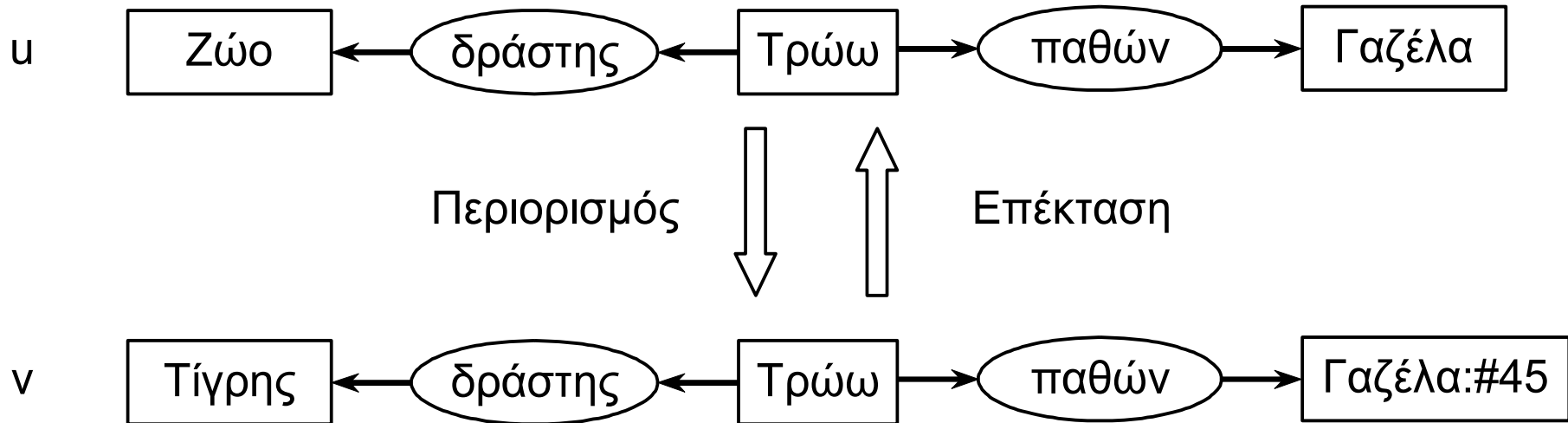
- ❖ είδος γραμματικής για ορθά δομημένους γράφους
- ❖ 6 κανόνες που παράγουν έναν ορθά δομημένο γράφο w επιδρώντας στους επίσης ορθά δομημένους γράφους u και v
- ❖ 3 κατηγορίες:
 - ❑ Κανόνες εξειδίκευσης (περιορισμός και συνένωση)
 - ❑ Κανόνες γενίκευσης (επέκταση και διαχωρισμούς)
 - ❑ Κανόνες ισοδυναμίας (αντιγραφή και απλοποίηση)
- ❖ Για κάθε κανόνα υπάρχει ο αντίστροφός του (που επιφέρει δηλαδή το αντίστροφο αποτέλεσμα)
 - ❑ Αντιγραφή - Απλοποίηση
 - ❑ Περιορισμός - Επέκταση
 - ❑ Συνένωση - Διαχωρισμός

Αντιγραφή - Απλοποίηση



- ❖ Πρόκειται για **κανόνες ισοδυναμίας**: οι δύο εκφράσεις είναι αληθείς ή ψευδείς υπό τις ίδιες ακριβώς συνθήκες.
- ❖ Η διακεκομμένη γραμμή αναπαριστά την **συναναφορά** (coreferent link).

Περιορισμός και Επέκταση



❖ *Περιορισμός μέσω του τύπου της έννοιας (restriction by type)*

□ Τίγρης \leq Ζώο.

❖ *Περιορισμός μέσω της αναφοράς (restriction by referent)*

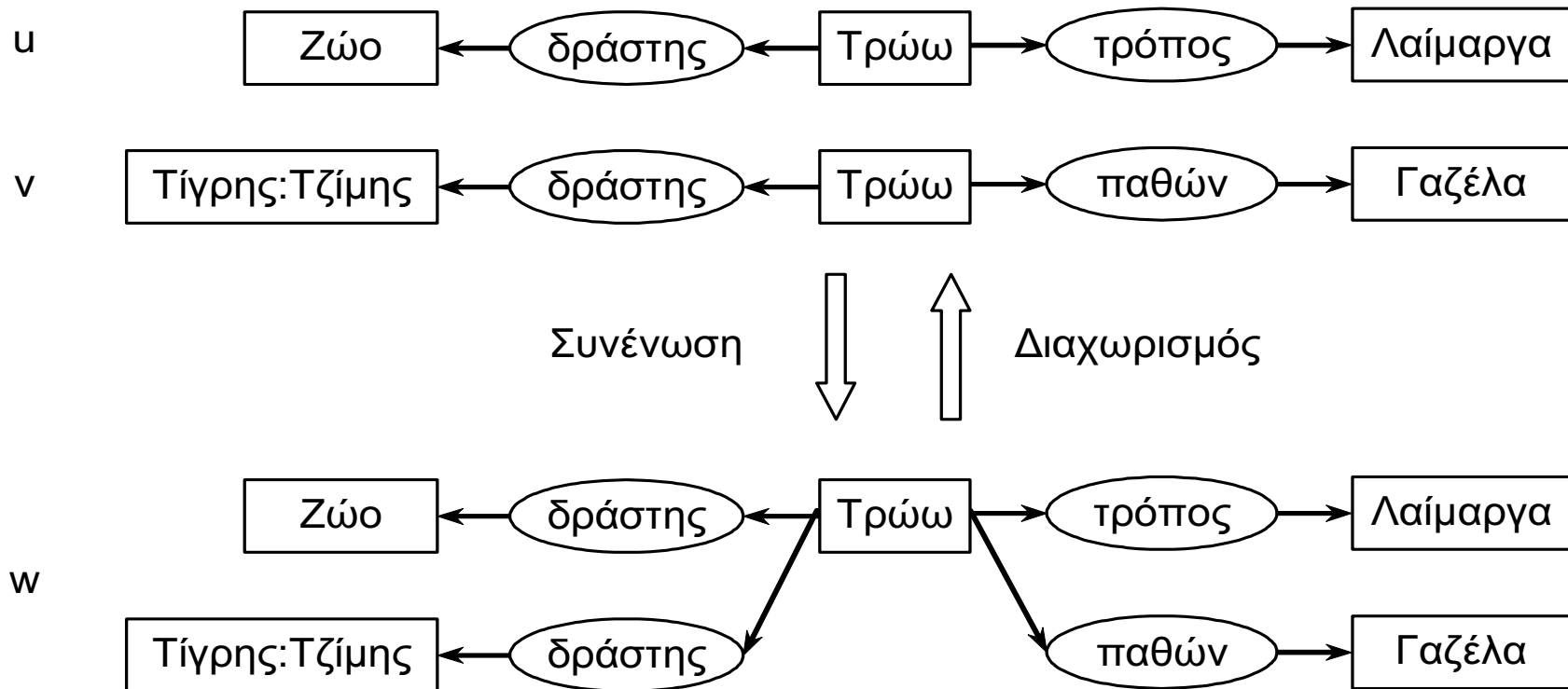
□ [Γαζέλα] \implies [Γαζέλα:#45]

❖ *Η επέκταση είναι ένας κανόνας γενίκευσης (generalization rule).*

□ αν "ένας τίγρης τρώει τη γαζέλα #45" (γράφος v) τότε οπωσδήποτε "κάποιο ζώο τρώει κάποια γαζέλα" (γράφος u).

❖ *Ο περιορισμός είναι κανόνας εξειδίκευσης (specialization rule).*

Συνένωση και Διαχωρισμός

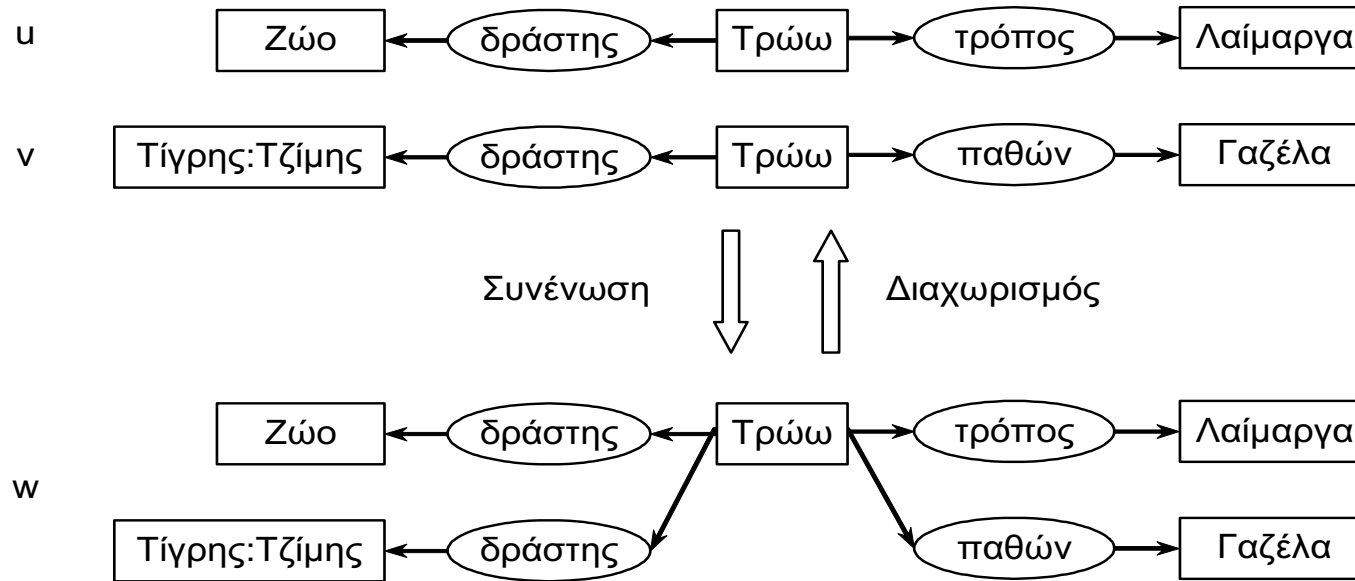


- ❖ Γίνεται πάνω στην κοινή έννοια με απαλειφή των πλεοναζόντων τόξων.
- ❖ Η συνένωση είναι κανόνας εξειδίκευσης (*specialization rule*).
- ❖ Ο διαχωρισμός είναι κανόνας γενίκευσης (*generalization rule*).

Κανόνες Ορθής Διαμόρφωσης (2/2)

❖ ΔΕΝ αποτελούν κανόνες εξαγωγής συμπερασμάτων.

❖ Παράδειγμα:



❑ u και v είναι αληθείς γράφοι

❑ ο προκύπτων γράφος w είναι εξασφαλισμένα ορθός δομικά

❑ Τι γίνεται με το νόημα;

- Οι κανόνες εξειδίκευσης δε διατηρούν την τιμή αληθείας των προτάσεων στις οποίες δρουν.
- Δεν είναι σίγουρο ότι ο τίγρης 'Τζίμης' είναι το ζώο που τρώει λαίμαργα!

- ❖ **Συμπέρασμα:** δεν επαρκούν για συλλογιστική με εννοιολογικούς γράφους.

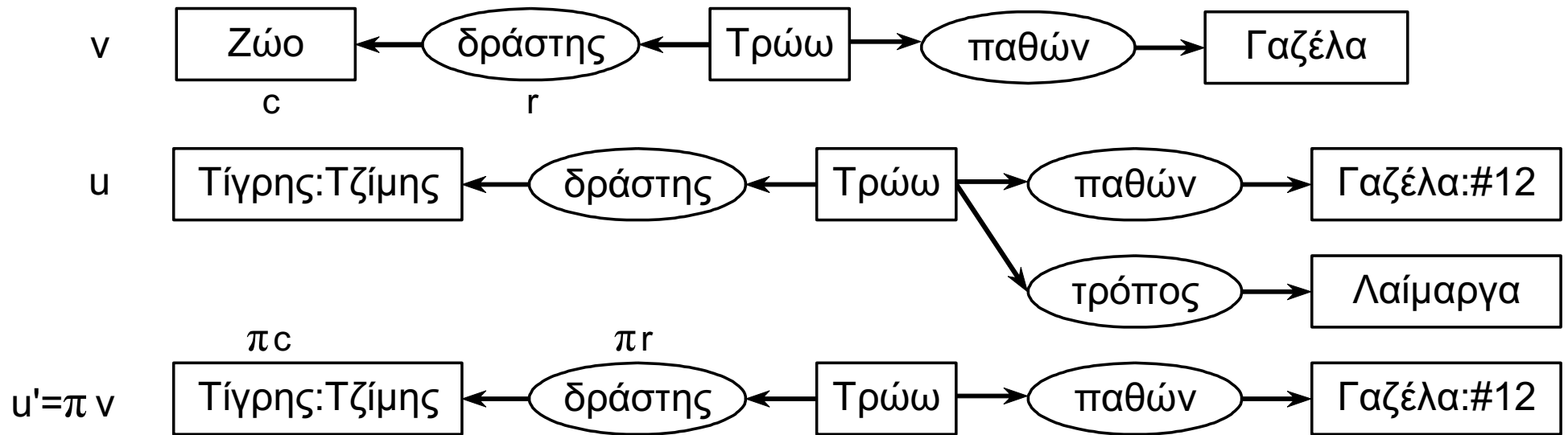
Προβολή

- ❖ Η χρήση των κανόνων ορθής διαμόρφωσης παράγει γενικεύσεις (generalizations) και εξειδικεύσεις (specializations).

- η γενίκευση ενός γράφου που είναι αληθής οδηγεί σε γράφο που επίσης είναι αληθής
- η εξειδίκευση ενός γράφου που είναι ψευδής οδηγεί σε γράφο που επίσης είναι ψευδής

- ❖ **προβολή (projection):**

- ο γράφος u' ονομάζεται **προβολή** του v στον u και συμβολίζεται με πv
- η όλη διαδικασία ονομάζεται επίσης "προβολή"
- Πρακτικά: ο γράφος v υπάρχει (όχι αυτούσιος) μέσα στον γράφο u



Παραγωγή Γνώσης

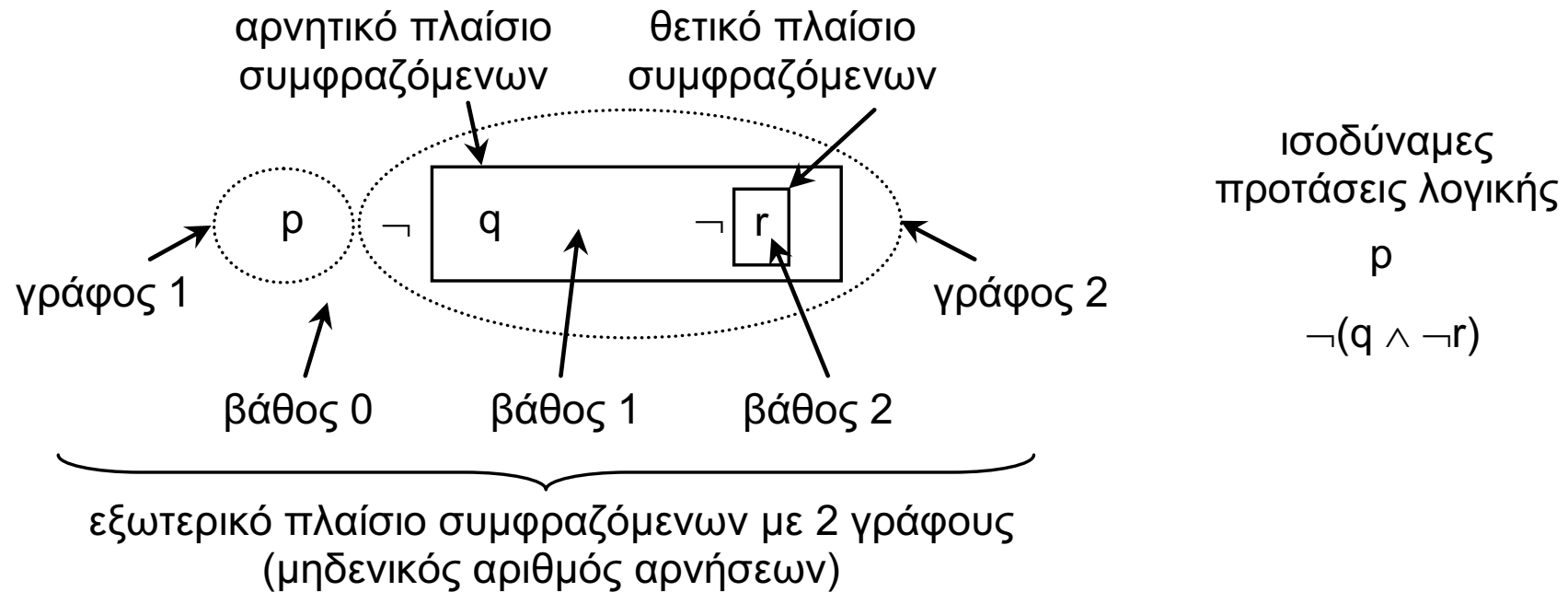
- ❖ Η διαδικασία **προβολής** αποτελεί μηχανισμό ενοποίησης (unification) σε εννοιολογικούς γράφους.
- ❖ Έστω μια βάση γνώσης με:
 - ❑ κανόνες if-then που έχουν εννοιολογικούς γράφους ως προϋπόθεση και συμπέρασμα
 - ❑ στατική γνώση κωδικοποιημένη σε εννοιολογικούς γράφους
- ❖ Είναι δυνατό να γίνει ενοποίηση κάποιου γράφου με την προϋπόθεση του κανόνα, οπότε συμπεραίνεται ο γράφος-συμπέρασμα αυτού.

□ παράγεται δηλαδή νέα γνώση

❖ Σε απλούς εννοιολογικούς γράφους (χωρίς έννοιες συμφραζομένων, αρνήσεις και συναναφορές) ο μηχανισμός της προβολής είναι sound και complete.

Κανόνες Εξαγωγής Συμπερασμάτων (1/3)

- ❖ Αποτελούν προσαρμογή (Sowa) αντίστοιχου συστήματος κανόνων κατηγορηματικού λογισμού που είχε προταθεί από τον Peirce στις αρχές του 19^{ου} αιώνα.
- ❖ **Θετικό** και **αρνητικό** πλαίσιο συμφραζόμενων.



- Ένα πλαίσιο συμφραζόμενων είναι **θετικό** (*positive context*) εάν βρίσκεται φωλιασμένο σε ζυγό αριθμό αρνήσεων (πιθανώς και σε μηδενικό αριθμό αρνήσεων).

- ❑ Ένα πλαίσιο συμφραζόμενων είναι *αρνητικό* (*negative context*) εάν βρίσκεται φωλιασμένο σε μονό αριθμό αρνήσεων.

Κανόνες Εξαγωγής Συμπερασμάτων (2/3)

- ❖ Οι κανόνες ισοδυναμίας παραμένουν ως έχουν σε οποιοδήποτε πλαίσιο συμφραζόμενων.
- ❖ Οι κανόνες εξειδίκευσης σε αρνητικό πλαίσιο συμφραζόμενων
 - ❑ μετατρέπονται σε κανόνες γενίκευσης, ενώ
 - ❑ σε θετικό πλαίσιο συμφραζόμενων παραμένουν κανόνες εξειδίκευσης.
- ❖ Οι κανόνες γενίκευσης
 - ❑ σε αρνητικό πλαίσιο συμφραζόμενων μετατρέπονται σε κανόνες εξειδίκευσης, ενώ
 - ❑ σε θετικό πλαίσιο συμφραζόμενων παραμένουν κανόνες γενίκευσης.
- ❖ Η διπλή άρνηση (δύο διαδοχικά αρνητικά πλαίσια συμφραζόμενων μεταξύ των οποίων δεν παρεμβάλλεται κάτι άλλο), αποτελεί κανόνα ισοδυναμίας.
 - ❑ Μπορεί να προστεθεί ή να αφαιρεθεί γύρω από οποιοδήποτε γράφο (ή τμήμα γράφου).

Κανόνες Εξαγωγής Συμπερασμάτων (3/3)

- ❖ Οι κανόνες συλλογιστικής του Pierce όπως υιοθετήθηκαν από τον Sowa ώστε να υλοποιούν κατηγορηματικό λογισμό 1^{ης} τάξης για εννοιολογικούς γράφους είναι:
 - ❑ *Διαγραφή (erasure)*: Σε ένα θετικό πλαίσιο συμφραζόμενων, οποιοσδήποτε γράφος (ή τμήμα γράφου) u μπορεί να αντικατασταθεί με μία γενίκευση του u . Ειδικότερα, ο γράφος u μπορεί να διαγραφεί καθώς αυτό ισοδυναμεί με αντικατάστασή του με τον κενό γράφο.
 - ❑ *Εισαγωγή (insertion)*: Σε ένα αρνητικό πλαίσιο συμφραζόμενων, οποιοσδήποτε γράφος (ή τμήμα γράφου) u μπορεί να αντικατασταθεί με μία εξειδίκευση του u . Ειδικότερα, μπορεί να εισαχθεί οποιοσδήποτε γράφος u καθώς αυτός μπορεί να θεωρηθεί εξειδίκευση του κενού γράφου.
 - ❑ *Επανάληψη (iteration)*: Εάν ένας γράφος (ή τμήμα γράφου) u βρίσκεται εντός ενός πλαισίου συμφραζόμενων C , τότε ένα αντίγραφο του u μπορεί να εισαχθεί εντός του C ή εντός οποιουδήποτε άλλου πλαισίου συμφραζόμενων βρίσκεται εντός του C .
 - ❑ *Απαλοιφή επανάληψης (deiteration)*: Οποιοσδήποτε γράφος (ή τμήμα γράφου) u που μπορεί να θεωρηθεί ότι έχει προέλθει με εφαρμογή του κανόνα της επανάληψης, μπορεί να διαγραφεί.
 - ❑ *Ισοδυναμία (equivalence)*: Οποιοσδήποτε κανόνας ισοδυναμίας (αντιγραφή, απλοποίηση, διπλή άρνηση) μπορεί να εφαρμοστεί πάνω σε οποιοδήποτε γράφο (ή τμήμα γράφου) u , σε οποιοδήποτε πλαίσιο συμφραζόμενων.
- ❖ Όλοι οι παραπάνω κανόνες μπορεί να αναπαρασταθούν γραφικά.

Παράδειγμα Συλλογιστικής (1/3)

❖ Έστω η ακόλουθη γνώση:

□ Κάποιος είναι Έλληνας πολίτης αν και μόνο αν είναι αληθής οποιαδήποτε από τις ακόλουθες συνθήκες:

7) Γεννήθηκε στην Ελλάδα.

8) Ένας από τους γονείς του είναι Έλληνας πολίτης.

9) Έχει πολιτογραφηθεί ως Έλληνας.

❖ Λόγω του "αν και μόνο αν" κάθε έκφραση θα εμφανιστεί ως προϋπόθεση κάποιας συνεπαγωγής αλλά και ως επακόλουθο κάποιας άλλης.

1. "Εάν γεννήθηκε στη Ελλάδα, τότε είναι Έλληνας πολίτης."

←[[Ατομο: *x] ← (αντικείμενο) ← [Γεννήθηκε] → (τόπος) → [Χώρα: 'Ελλάδα']]
←[[Πολίτης: ?x] ← (μέλος) ← [Χώρα: 'Ελλάδα']]]

2. "Εάν είναι παιδί Έλληνα πολίτη, τότε είναι Έλληνας πολίτης"

←[[Ατομο: *x] ← (παιδί) ← [Πολίτης] ← (μέλος) ← [Χώρα: 'Ελλάδα']]
←[[Πολίτης: ?x] ← (μέλος) ← [Χώρα: 'Ελλάδα']]]

3. "Εάν πολιτογραφήθηκε στην Ελλάδα, τότε είναι Έλληνας πολίτης"

←[[Ατομο: *x] ← (αποδέκτης) ← [Πολιτογράφηση] → (τοποθεσία) → [Χώρα: 'Ελλάδα']]

←[[Πολίτης: ?x] ← (μέλος) ← [Χώρα: 'Ελλάδα']]
]

Παράδειγμα Συλλογιστικής (2/3)

4. "Είναι Έλληνας πολίτης εάν: είτε γεννήθηκε στην Ελλάδα, ή είναι παιδί Έλληνα πολίτη, ή πολιτογραφήθηκε στην Ελλάδα"

←[[Πολίτης: *x] ← (μέλος) ← [Χώρα: 'Ελλάδα']]

←[[Άτομο: ?x] ← (αντικείμενο) ← [Γεννήθηκε] → (τόπος) → [Χώρα: 'Ελλάδα']]

]

←[[Άτομο: ?x] ← (παιδί) ← [Πολίτης] ← (μέλος) ← [Χώρα: 'Ελλάδα']]]

←[

[Άτομο: ?x] ← (αποδέκτης) ← [Πολιτογράφιση] → (τοποθεσία) → [Χώρα: 'Ελλάδα']]]

]

❖ Έστω ότι στους παραπάνω τέσσερις γράφους προστίθεται και ο γράφος:

5.

[Άτομο: 'John'] ← (αποδέκτης) ← [Πολιτογράφιση] → (τοποθεσία) → [Χώρα: 'Ελλάδα']]

❖ Μέσω του κανόνα της επανάληψης (iteration) ένα αντίγραφο του γράφου (5) μπορεί να εισαχθεί στο εξωτερικό αρνητικό πλαίσιο συμφραζόμενων του γράφου (3).

Προκύπτει έτσι ο γράφος (6):

6. ← [

[Άτομο: 'John'] ← (αποδέκτης) ← [Πολιτογράφηση] → (τοποθεσία) → [Χώρα: 'Ελλάδα']

[Άτομο: *x] ← (αποδέκτης) ← [Πολιτογράφηση] → (τοποθεσία) → [Χώρα: 'Ελλάδα']

← [[Πολίτης: ?x] ← (μέλος) ← [Χώρα: 'Ελλάδα']]

]

Παράδειγμα Συλλογιστικής (3/3)

❖ Στο γράφο (6), οι δύο πρώτοι εσωτερικοί γράφοι (βάθος 1), μπορεί να συνενωθούν

7. $\leftarrow [$

$[\text{Άτομο} : ' \text{John} '] \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα} : ' \text{Ελλάδα} ']]$
 $\leftarrow [[\text{Πολίτης} : ' \text{John} '] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα} : ' \text{Ελλάδα} ']]]$
]

❖ Μέσω του κανόνα της απαλοιφής επανάληψης (deiteration) ο πρώτος εσωτερικός γράφος στο γράφο (7), μπορεί να απαλειφθεί καθώς είναι ακριβές αντίγραφο του γράφου (5) που εξακολουθεί να υπάρχει στο αρχικό σύνολο γράφων.

8. $\leftarrow [\leftarrow [[\text{Πολίτης} : ' \text{John} '] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα} : ' \text{Ελλάδα} ']]]]$

❖ Η διπλή άρνηση που ορίζουν τα δύο διαδοχικά φωλιασμένα αρνητικά πλαίσια συμφραζόμενων στο γράφο (8), μπορεί να απαλειφθεί, οπότε προκύπτει το συμπέρασμα ότι "Ο John είναι Έλληνας πολίτης" (γράφος (9)).

9. $[\text{Πολίτης} : ' \text{John} '] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα} : ' \text{Ελλάδα} ']]$

Αναπαράσταση με Κανόνες

- ❖ Πολύ πρακτικός τρόπος αναπαράστασης για την εξαγωγή συμπερασμάτων
- ❖ Αποτελούν τη βάση πολλών συστημάτων γνώσης (knowledge systems)
- ❖ Γενικά Πλεονεκτήματα:
 - ❑ Κοντά στην ανθρώπινη γνώση
 - ❑ Επάρκεια συνεπαγωγών
- ❖ Συγκεκριμένα Πλεονεκτήματα:
 - ❑ *Modularity*: Κάθε κανόνας ορίζει ένα μικρό, (σχεδόν) ανεξάρτητο τμήμα της γνώσης
 - ❑ *Incrementability*: Μπορούν να προστεθούν νέοι κανόνες (σχεδόν) ανεξάρτητα από τους υπάρχοντες
 - ❑ *Modifiability*: Οι υπάρχοντες κανόνες μπορούν να αλλάξουν (σχεδόν) ανεξάρτητα από τους υπόλοιπους

Είδη Κανόνων

Μορφές Κανόνων	Εκφράζει	Επεξήγηση
IF συνθήκες THEN συμπέρασμα Συνεπαγωγικός (Deductive) κανόνας	Δηλωτική γνώση	Αν οι συνθήκες αληθεύουν τότε αληθεύει και το συμπέρασμα
IF συνθήκες THEN ενέργειες Κανόνας Παραγωγής (Production)	Διαδικαστική γνώση	Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες
ON συμβάν IF συνθήκες THEN ενέργειες Ενεργός (active) κανόνας	Διαδικαστική γνώση	Όταν συμβεί το γεγονός (συμβάν) Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες

❖ *Συνθήκες (conditions)*: ακολουθία από κατηγορήματα που συνδέονται με λογικούς τελεστές AND, OR

□ Προϋποθέσεις (premises) ή αριστερό μέρος του κανόνα (left hand side - LHS)

❖ *Συμπέρασμα (conclusion)*: κατηγορημα

❖ *Ενέργειες (actions)*: μία σειρά από εντολ που πρέπει να εκτελεστούν

Επακόλουθα (consequent) ή δεξιό μέρος του κανόνα (right hand side - RHS)

Συστήματα Κανόνων

- ❖ Συστήματα εξαγωγής συμπερασμάτων (deduction systems)
 - ❑ π.χ. Prolog, Datalog
 - ❑ Γνώση που δηλώνει μία αλήθεια για τον κόσμο, αλλά δεν αναφέρει ρητά πότε και πώς εφαρμόζεται
- ❖ Συστήματα παραγωγής (production systems)
 - ❑ π.χ. CLIPS, Flex
 - ❑ Γνώση για το ποιες συγκεκριμένες ενέργειες πρέπει να εκτελεστούν δεδομένης μιας κατάστασης
 - ❑ Οι εκτελούμενες ενέργειες επιφέρουν μη-αναστρέψιμα αποτελέσματα
- ❖ Ενεργά Συστήματα ((re-)active systems, active database systems)
 - ❑ π.χ. Oracle Triggers, Δαίμονες πλαισίων Flex

Ενεργοί Κανόνες

- ❖ Οι κανόνες παραγωγής δηλώνουν διαδικαστική γνώση
- ❑ Οι κανόνες εκτελούνται "όταν η συνθήκη είναι αληθής"
- ❑ Δεν είναι σαφώς ορισμένο πότε ακριβώς εκτελούνται οι ενέργειες
- ❑ Αν και εκφράζουν διαδικαστική γνώση, η συνθήκη τους περιέχει δηλωτική γνώση.
- ❖ Οι ενεργοί κανόνες (active rules) εκφράζουν *καθαρά διαδικαστική γνώση*
- ❑ Κανόνες οδηγούμενοι από συμβάντα ή γεγονότα (event-driven rules)
- ❖ Εκφράζουν με σαφήνεια το πότε ακριβώς ενεργοποιούνται:
- ❑ Όταν συμβεί ένα συγκεκριμένο συμβάν
- ❑ Τότε και μόνο τότε εξετάζεται η συνθήκη τους και αν ικανοποιείται, τότε εκτελούνται οι ενέργειές τους.
- ❖ Παραδείγματα συμβάντων:
- ❑ Μία συγκεκριμένη χρονική στιγμή του ρολογιού του συστήματος
- ❑ Ένα πάτημα του αριστερού πλήκτρου του ποντικού ή ενός πλήκτρου του πληκτρολογίου
- ❑ Η επιλογή κάποιου μενού από το χρήστη
- ❑ Η προσπάθεια προσπέλασης ή αλλαγής κάποιων "ευαίσθητων" δεδομένων, κλπ.

Παράδειγμα Αναπαράστασης με Κανόνες

Σύμπτωμα	Πιθανή Βλάβη	Επιδιόρθωση
Ο εκτυπωτής τυπώνει σωστά αλλά τα χρώματα δε τυπώνονται σωστά	Έχει τελειώσει το έγχρωμο μελάνι	Αλλάξτε την κεφαλή με το έγχρωμο μελάνι

❖ Συνεπαγωγικός Κανόνας

IF ο εκτυπωτής τυπώνει σωστά **AND**
τα χρώματα δε τυπώνονται σωστά
THEN έχει τελειώσει το έγχρωμο μελάνι

❖ Κανόνας Παραγωγής

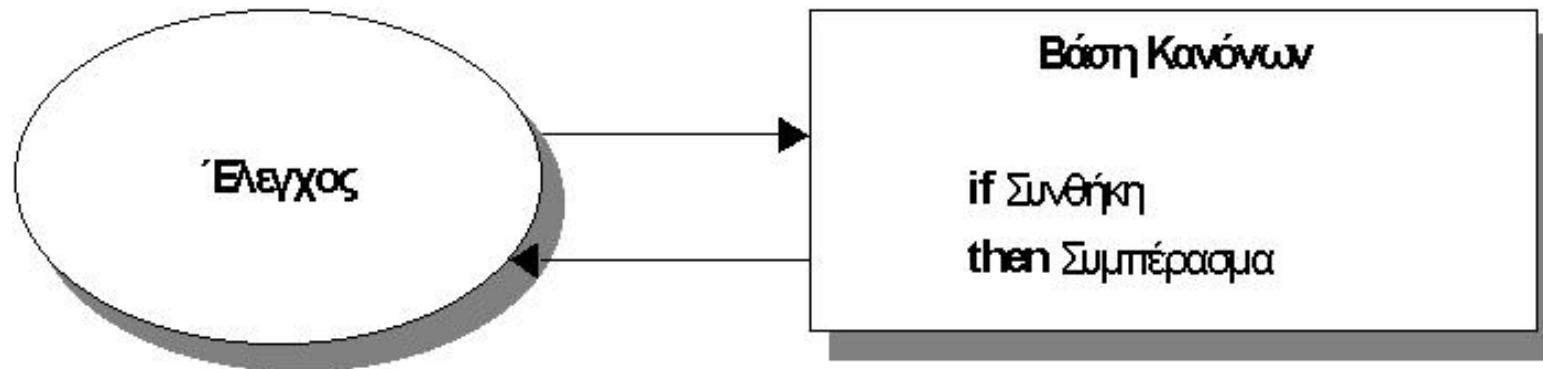
IF ο εκτυπωτής τυπώνει σωστά **AND**
τα χρώματα δε τυπώνονται σωστά
THEN αλλάξτε την κεφαλή με το έγχρωμο μελάνι

❖ Ενεργός Κανόνας

ON εκτύπωση
IF τα χρώματα δε τυπώνονται σωστά
THEN αλλάξτε την κεφαλή με το έγχρωμο μελάνι

Συστήματα Εξαγωγής Συμπερασμάτων

Deduction Systems



- ❖ **Βάση κανόνων (rule base):** περιέχει ένα σύνολο από κανόνες
- ❖ **Έλεγχος (control):** καθορίζει τον τρόπο εκτέλεσης των κανόνων για την εξαγωγή συμπερασμάτων
 - ❑ Ποιοι κανόνες είναι υποψήφιοι για να επιλύσουν το πρόβλημα;
 - ❑ Με ποιόν τρόπο θα γίνει η επιλογή;
 - ❑ Ποιος από τους κανόνες τελικά θα επιλεγεί;
 - ❑ Τι θα γίνει με τους υπόλοιπους κανόνες;

Εξαγωγή Συμπερασμάτων

- ❖ Χρησιμοποιείται η συνεπαγωγική συλλογιστική.
- ❑ Υλοποιείται από την *ακολουθία εκτέλεσης κανόνων (chaining)*
- ❑ Αλγόριθμος που συνδυάζει τα δεδομένα, τους κανόνες και τα ενδιάμεσα συμπεράσματα
- ❖ *Ορθή ακολουθία εκτέλεσης (forward chaining)*
- ❑ Εξετάζονται πρώτα αν οι προϋποθέσεις στο αριστερό μέρος του κανόνα είναι αληθείς έτσι ώστε το συμπέρασμα που αναφέρεται στο δεξιό μέρος να είναι αληθές.
- ❑ Εξετάζονται μόνο οι αληθείς τρόποι απόδειξης
- ❑ Μπορεί να συμπεράνει περισσότερα συμπεράσματα από τα επιθυμητά
- ❑ Ενδείκνυται όταν υπάρχουν λίγα δεδομένα που οδηγούν σε πολλά συμπεράσματα
- ❑ **Εφαρμογές:** Συστήματα Διάγνωσης, Συστήματα Παραγωγής
- ❖ *Ανάστροφη ακολουθία εκτέλεσης (backward chaining)*
- ❑ Ξεκινά από το δεξιό μέρος του κανόνα και προσπαθεί να βρει αν οι προϋποθέσεις είναι αληθείς
- ❑ Εξετάζονται όλοι οι εναλλακτικοί τρόποι απόδειξης του συμπεράσματος (ακόμα και οι μη-αληθείς) έως ότου αποδειχθεί η αλήθεια του συμπεράσματος (π.χ. Prolog).
- ❑ Ενδείκνυται όταν υπάρχουν λίγα συμπεράσματα και πολλά δεδομένα
- ❑ Το σύστημα ζητά τα δεδομένα με λογική σειρά και μόνο όσα χρειάζονται

- **Εφαρμογές:** Συστήματα Ελέγχου Λειτουργίας (Monitoring)

Αναπαράσταση με Κανόνες

Παράδειγμα

```
1:  if  has(Animal,hair)                or 5:  if      isa(Animal,carnivore) and
gives(Animal,milk)                      has(Animal,tawny_colour) and
then isa(Animal,mammal).                has(Animal,black_stripes)
                                         then   isa(Animal,tiger).

2:  if  has(Animal,feathers) or
      (flies(Animal) and lays(Animal,eggs))
then   isa(Animal,bird).

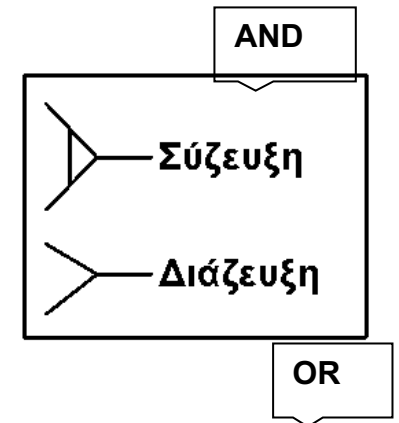
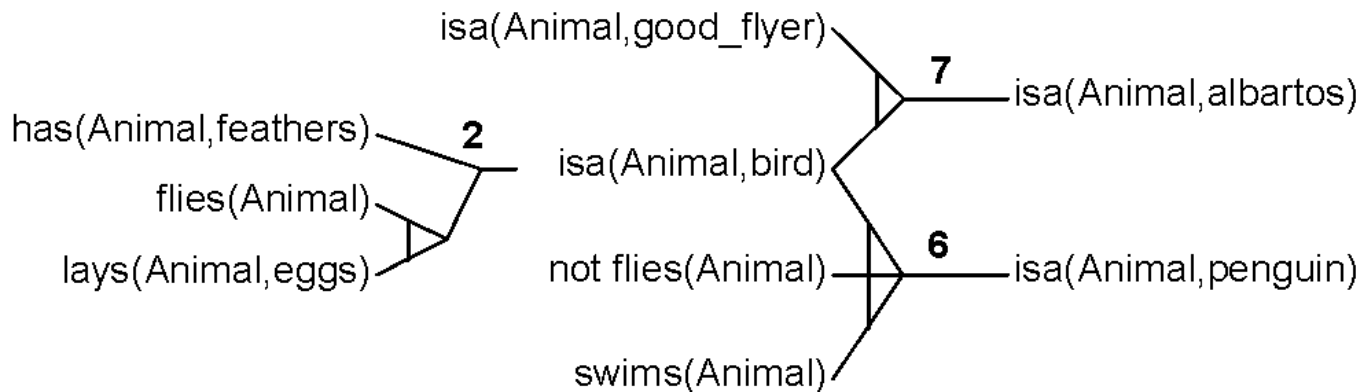
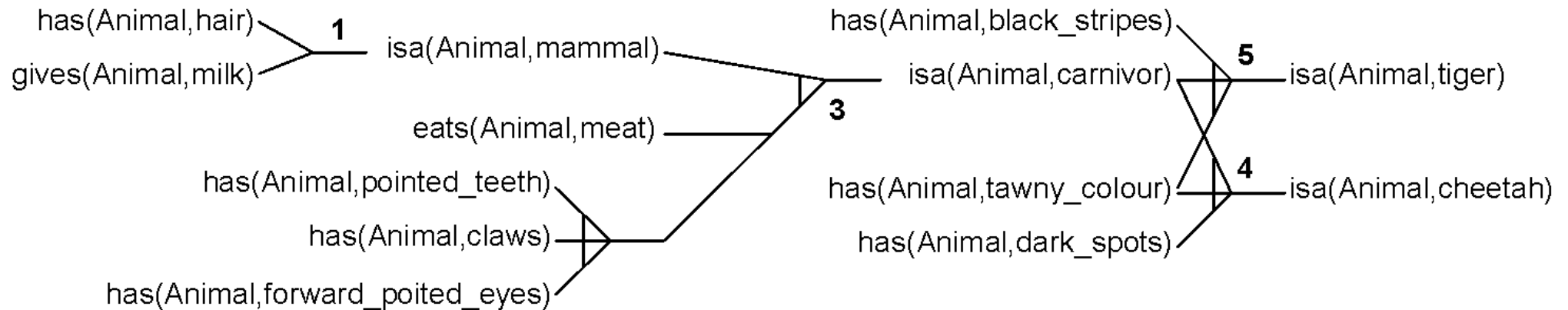
3:  if   isa(Animal,mammal) and
      (eats(Animal,meat) or
      (has(Animal,pointed_teeth) and
      has(Animal,claws) and
      has(Animal,forward_pointing_eyes)))
then   isa(Animal,carnivore).

4:  if  isa(Animal,carnivore) and
      has(Animal,tawny_colour) and
      has(Animal,dark_spots)
then   isa(Animal,cheetah).

6:  if  isa(Animal,bird) and
      not flies(Animal) and
      swims(Animal)
then   isa(Animal,penguin).

7:  if  isa(Animal,bird) and
      isa(Animal,good_flyer)
then   isa(Animal,albatros).
```


Γραφική Αναπαράσταση Κανόνων

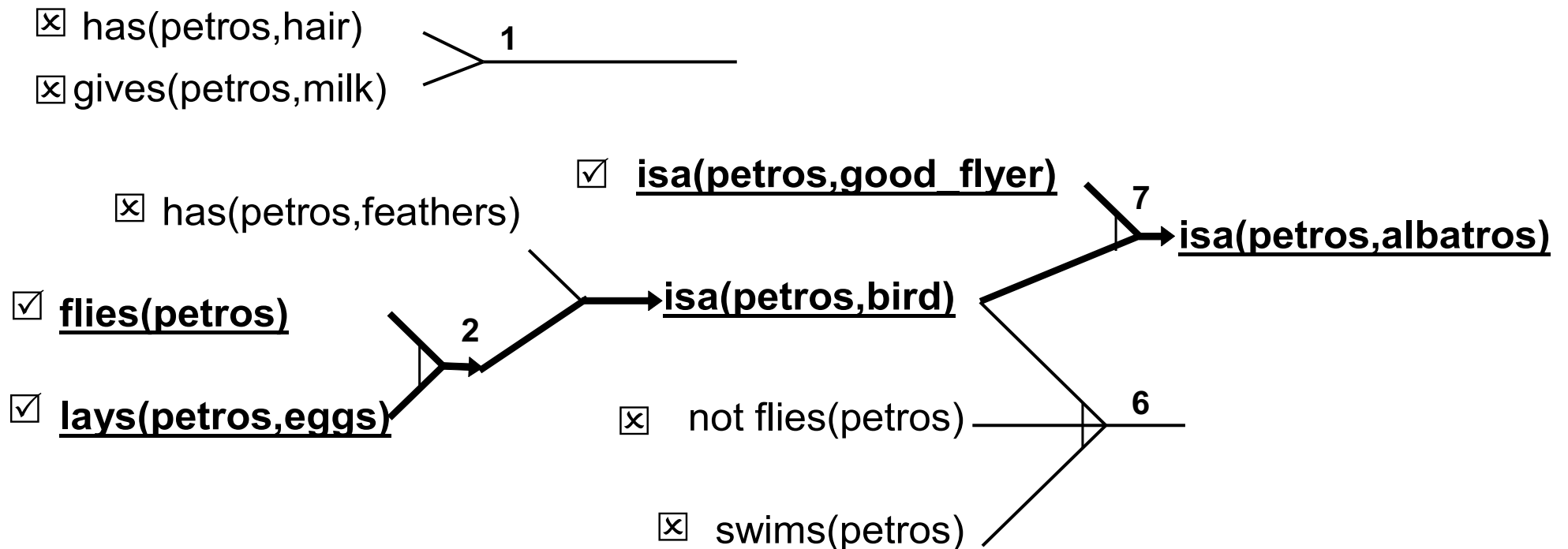


❖ Μπορεί να χρησιμοποιηθεί ορθή ή ανάστροφη ακολουθία εκτέλεσης

Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

Ορθή Ακολουθία Εκτέλεσης

❖ Αρχικά δεδομένα: **flies(petros)**, **lays(petros, eggs)**, **isa(petros, goodflyer)**.

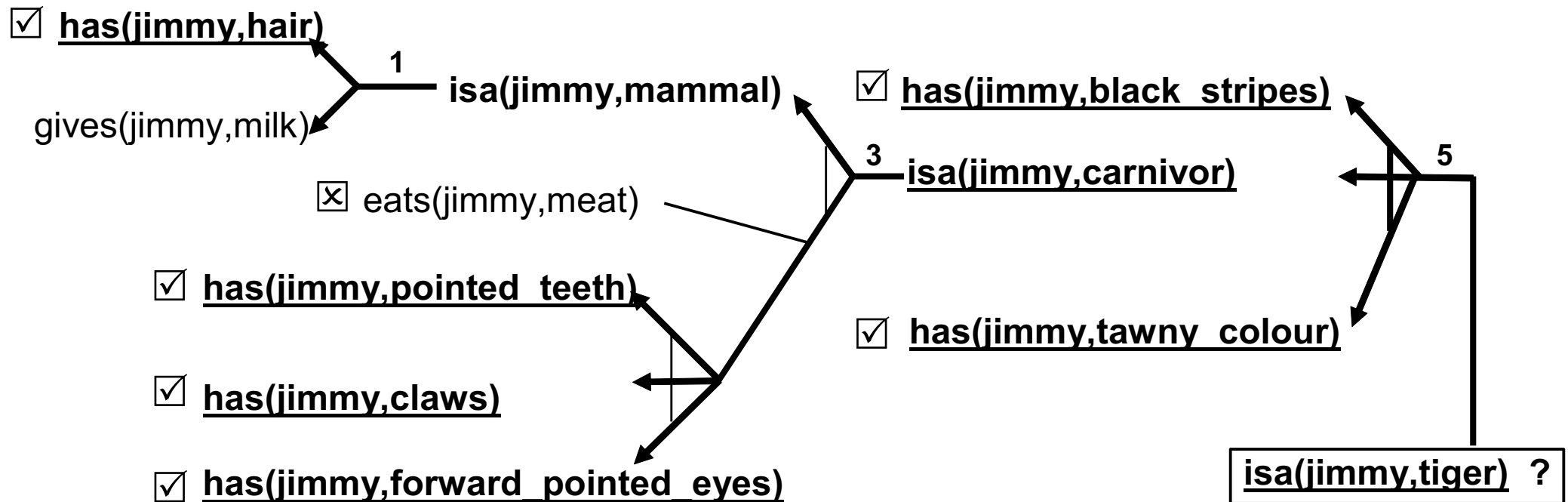


❖ Παράγονται τα συμπεράσματα: **isa(petros, albatros)** και **isa(petros, bird)**

Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

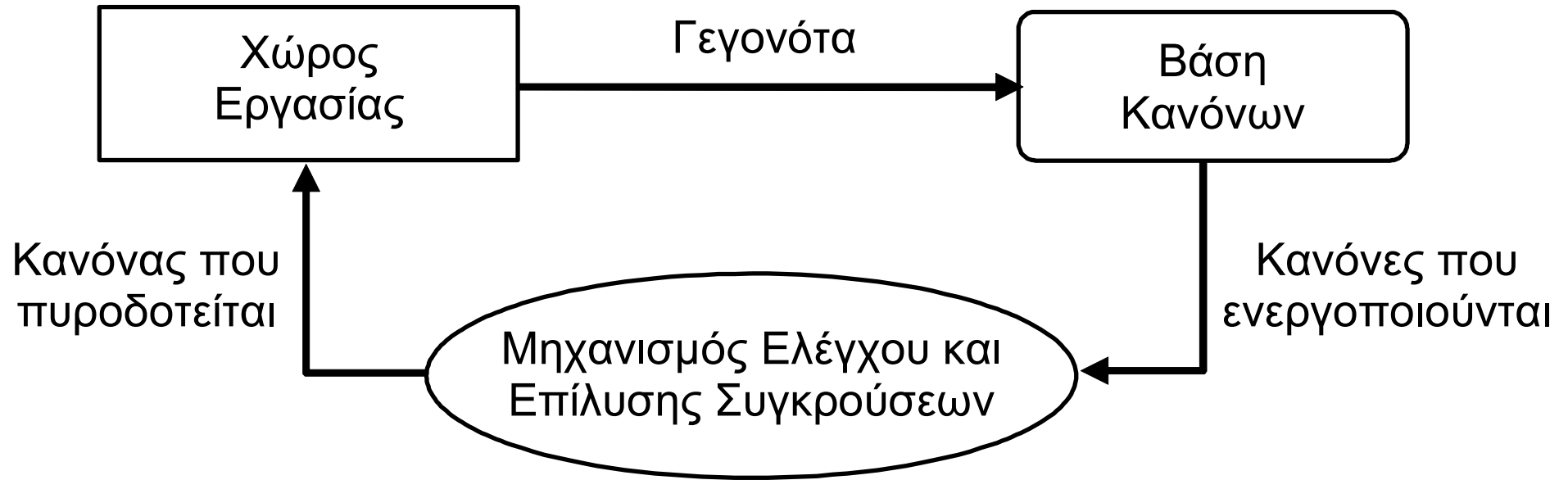
Ανάστροφη Ακολουθία Εκτέλεσης

❖ Ερώτηση: **isa(jimmy, tiger)**



❖ Απάντηση: **Yes**

Συστήματα Παραγωγής



- ❖ Βάση κανόνων: περιέχει τους κανόνες παραγωγής
- ❖ Χώρος εργασίας (working memory): περιέχει γεγονότα
 - ❑ Αρχικά δεδομένα (data) ή ενδιάμεσα συμπεράσματα (partial conclusions)
 - ❑ Στοιχεία της μνήμης εργασίας (working memory elements)
- ❖ Μηχανισμός ελέγχου (control ή scheduler) και επίλυσης συγκρούσεων (conflict resolution): εκτέλεση των κανόνων βάσει στρατηγικής επίλυσης συγκρούσεων (conflict resolution strategy)

Αβέβαιη Γνώση

- ❖ Κυριότερες πηγές αβεβαιότητας:
 - ❑ Ανακριβή δεδομένα (*imprecise data*).
 - ❑ Ελλιπή δεδομένα (*incomplete data*)
 - ❑ Υποκειμενικότητα ή/και ελλείψεις στην περιγραφή της γνώσης
 - ❑ Κάθε είδους **περιορισμοί** που κάνουν το όλο πλαίσιο λήψης απόφασης ατελές.

- ❖ Ανάγκη ύπαρξης "μη ακριβών" μεθόδων συλλογισμού.
 - ❑ Θεωρία Πιθανοτήτων
 - ❑ Συντελεστές Βεβαιότητας (*Certainty Factors*)
 - ❑ Θεωρία Dempster-Shafer
 - ❑ Ασαφής Λογική (*Fuzzy Logic*).

Θεωρία Πιθανοτήτων

❖ Αν E είναι ένα γεγονός, η *άνευ συνθηκών πιθανότητα* (*unconditional probability*) $P(E)$ να συμβεί το γεγονός εκφράζεται με έναν πραγματικό αριθμό για τον οποίο ισχύουν:

- ❑ $0 \leq P(E) \leq 1$
- ❑ $P(E) = 1$ αν E σίγουρο γεγονός
- ❑ $P(E) + P(\leftarrow E) = 1$

❖ *Πιθανότητα υπό συνθήκη* (*conditional probability*):

❑ Η πιθανότητα να ισχύει το υποθετικό συμπέρασμα H δεδομένης της ισχύος μόνο του γεγονότος E .

$$P(H | E) = \frac{P(H \wedge E)}{P(E)}$$

❖ *Ιδιότητες*

- ❑ *Προσθετική Ιδιότητα:* $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- ❑ *Πολ/στική Ιδιότητα για δύο ανεξάρτητα γεγονότα A και B :* $P(A \wedge B) = P(A) \cdot P(B)$
- ❑ *Πολ/στική Ιδιότητα για δύο μη ανεξάρτητα γεγονότα A και B :* $P(A \wedge B) = P(A) \cdot P(B|A)$

Παράδειγμα

❖ Έστω ότι έχουμε ένα ζάρι:

❖ $P(A) = P(\text{περιττός αριθμός}) = 3/6 = 0.5$

□ γιατί υπάρχουν 3 δυνατές τιμές (1,3,5) από σύνολο 6 δυνατών τιμών (1,2,3,4,5,6)

❖ $P(B) = P(\text{αριθμός} \leq 3) = 3/6 = 0.5$

□ γιατί υπάρχουν 3 δυνατές τιμές (1,2,3) από σύνολο 6 δυνατών τιμών (1,2,3,4,5,6)

❖ $P(B|A) = P(\text{αριθμός} \leq 3 \text{ δεδομένου ότι είναι περιττός}) = 2/3$

□ γιατί υπάρχουν 2 δυνατές τιμές (1,3) από σύνολο 3 δυνατών τιμών (1,3,5)

❖ $P(A \cap B) = P(\text{περιττός αριθμός και} \leq 3) = P(A) * P(B|A) = 0.5 * 2/3 = 0.33$

❖ $P(A \cup B) = P(\text{περιττός ή} \leq 3) = P(A) + P(B) - P(A \cap B) = 0.5 + 0.5 - 0.33 = 0.67$
(προσθετική ιδιότητα)

Ο Νόμος του Bayes (*Bayes' rule*)

❖ Επιτρέπει τον υπολογισμό πιθανοτήτων υπό συνθήκη με χρήση άλλων πιθανοτήτων που είναι ευκολότερο να υπολογιστούν.

❖ Χρήση εκτιμήσεων αντί συχνοτήτων εμφάνισης γεγονότων.

❖ Η απλούστερη εκδοχή του νόμου του Bayes:
$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)}$$

□ Πιο εύκολο να χρησιμοποιηθεί, συγκριτικά με την σχέση της πιθανότητας υπό συνθήκη.

□ Αν Η μία ασθένεια και E ένα σύμπτωμα που σχετίζεται με αυτήν, τότε για τον υπολογισμό της πιθανότητας υπό συνθήκη απαιτείται πληροφορία που συνήθως δεν είναι διαθέσιμη:

- Πόσοι άνθρωποι στον κόσμο πάσχουν από την Η και ταυτόχρονα εμφανίζουν το σύμπτωμα E.
- Πόσοι εμφανίζουν απλά το σύμπτωμα E.

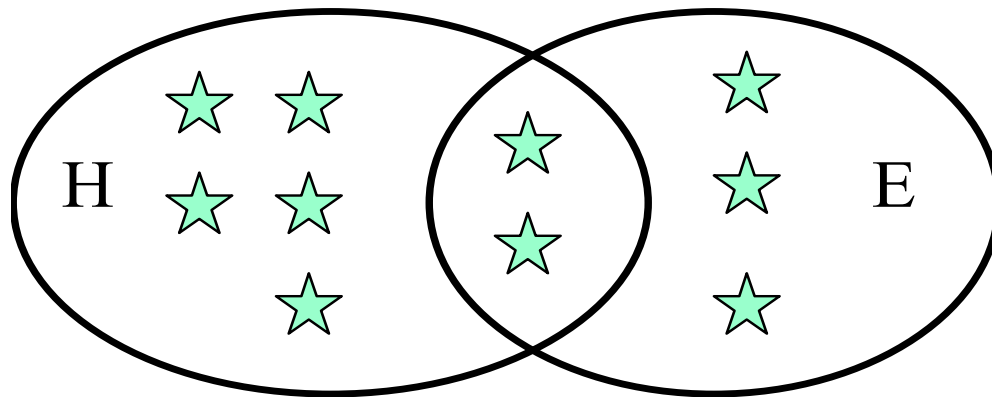
□ Στο νόμο του Bayes:

• Ένας γιατρός μπορεί να δώσει μία εκτίμηση για το πόσοι ασθενείς που έπασχαν από την ασθένεια Η εμφάνιζαν το σύμπτωμα E (ποσότητα $P(E|H)$). Αντίθετα, το κλάσμα των ασθενών με σύμπτωμα E που πάσχουν από την ασθένεια Η, δηλαδή ο όρος $P(H|E)$, τις περισσότερες φορές είναι αδύνατο να εκτιμηθεί.

- Το $P(H)$ μπορεί να υπολογιστεί από στατιστικά στοιχεία για τον συνολικό πληθυσμό.
- Το $P(E)$ από στατιστικά στοιχεία του ίδιου του γιατρού.

Παράδειγμα 1

- ❖ Έστω τα δύο σύνολα H και E με επτά και πέντε γεγονότα αντίστοιχα από ένα συνολικό πληθυσμό δέκα γεγονότων.
- ❖ Το σχήμα μας επιτρέπει να υπολογίσουμε τις απλές (άνευ συνθήκης) και τις υπό συνθήκη πιθανότητες με απλή εφαρμογή του ορισμού τους.



$$\begin{aligned}P(E) &= 5/10 = 0.5 \\P(H) &= 7/10 = 0.7 \\P(H|E) &= 2/5 = 0.4 \\P(E|H) &= 2/7 = 0.287514\end{aligned}$$

- ❖ Στο παράδειγμα: $P(H|E) * P(E) = P(E|H)*P(H)$
- ❖ Γνωρίζοντας τρεις από τις πιθανότητες μπορούμε να υπολογίσουμε την τέταρτη.

Γενική Σχέση του Νόμου του Bayes

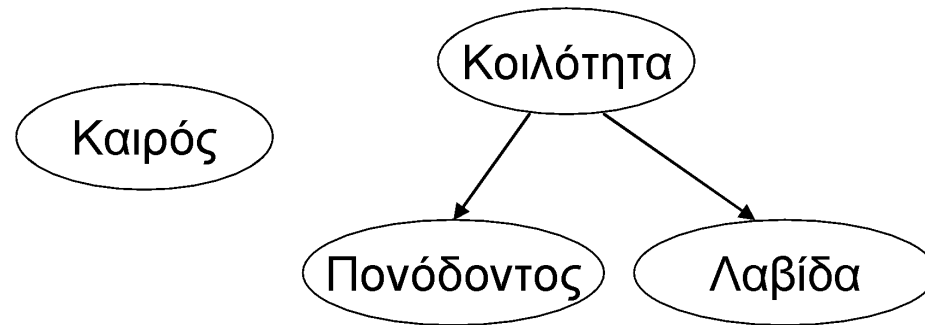
- ❖ Η πιθανότητα να ισχύει το υποθετικό συμπέρασμα H δεδομένης της ισχύος των γεγονότων E_1, E_2, \dots, E_k :

$$P(H | E_1 \wedge E_2 \wedge \dots \wedge E_k) = \frac{P(E_1 \wedge E_2 \wedge \dots \wedge E_k | H) \cdot P(H)}{P(E_1 \wedge E_2 \wedge \dots \wedge E_k)}$$

- ❖ **Πρόβλημα χρήσης:** για m πιθανές ασθένειες και n δυνατά συμπτώματα από τα οποία εμφανίζονται τα k , απαιτούνται $(m \cdot n)^k + m + n^k$ τιμές πιθανοτήτων, αριθμός υπερβολικά μεγάλος.
- ❖ **Απλούστερη περίπτωση:** αν τα διάφορα γεγονότα E θεωρούνται ανεξάρτητα το ένα από το άλλο, τότε απαιτούνται μόνο $m \cdot n + m + n$ τιμές πιθανοτήτων.
- ❖ Παράδειγμα: Τρεις Boolean μεταβλητές: Πονόδοντος, Κοιλότητα και Λαβίδα.
 - Η πλήρης συνδυασμένη κατανομή είναι ένας πίνακας $2^3=8$ γραμμών.
 - Αν συμπεριλάβουμε και τη μεταβλητή Καιρός που μπορεί να πάρει 4 τιμές (κρύος, με σύννεφα, βροχερός, αίθριος), η πλήρης κατανομή έχει 32 γραμμές.

- Από πολ/κή ιδιότητα έχουμε:
 - $P(\text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα} \wedge \text{Καιρός}) = P(\text{Καιρός} \mid \text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα}) * P(\text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα})$
- Για να απλοποιήσουμε το πρόβλημα σκεφτόμαστε ποιές μεταβλητές είναι ανεξάρτητες:
 - Φαίνεται λογικό ότι ο καιρός δεν επηρεάζεται από τα προβλήματα που έχει κάποιος στα δόντια του!
 - $P(\text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα} \wedge \text{Καιρός}) = P(\text{Καιρός}) * P(\text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα})$
 - Και έτσι έχουμε ένα πίνακα 4 γραμμών και ένα δεύτερο 8 γραμμών.
- Θα ήταν βολικό επίσης εάν οι μεταβλητές Πονόδοντος και Λαβίδα ήταν ανεξάρτητες, όμως δεν είναι: εάν η λαβίδα συναντήσει πιάσιμο στο δόντι αυτό σημαίνει ότι το δόντι πιθανώς έχει κοιλότητα η οποία με τη σειρά της προκαλεί πονόδοντο.
 - Αυτές οι μεταβλητές όμως είναι ανεξάρτητες με δεδομένη την παρουσία ή την απουσία της κοιλότητας (υπό συνθήκη ανεξαρτησία). Μαθηματικά αυτό γράφεται:
 - $P(\text{Πονόδοντος} \wedge \text{Λαβίδα} \mid \text{Κοιλότητα}) = P(\text{Πονόδοντος} \mid \text{Κοιλότητα}) * P(\text{Λαβίδα} \mid \text{Κοιλότητα})$
- Συνολικά έχουμε:

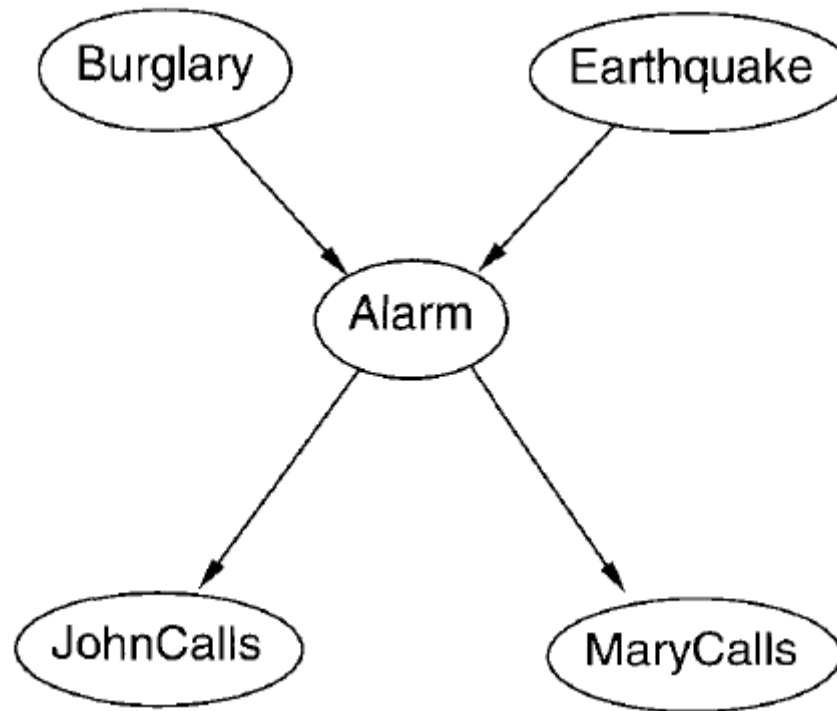
- $P(\text{Πονόδοντος} \wedge \text{Κοιλότητα} \wedge \text{Λαβίδα}) = P(\text{Πονόδοντος} \mid \text{Κοιλότητα}) * P(\text{Λαβίδα} \mid \text{Κοιλότητα}) * P(\text{Κοιλότητα})$



- Το οδοντιατρικό παράδειγμα απεικονίζει ένα συχνά εμφανιζόμενο μοτίβο στο οποίο μία μόνο αιτία επηρεάζει άμεσα πολλές επιδράσεις που όλες τους είναι υπό συνθήκη ανεξάρτητες με δεδομένη την αιτία αυτή (**Naive Bayes model**).

$$P(\text{Αιτία}, \text{Επίδραση}_1, \dots, \text{Επίδραση}_N) = P(\text{Αιτία}) \prod_{i=1}^N P(\text{Επίδραση}_i \mid \text{Αιτία})$$

Δίκτυα Πιθανοτήτων Bayes



- ❖ You have a new burglar alarm installed at home. It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes.
- ❖ You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm. John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too. Mary, on the other hand, likes rather loud music and sometimes misses the alarm altogether.

Δίκτυα Πιθανοτήτων Bayes

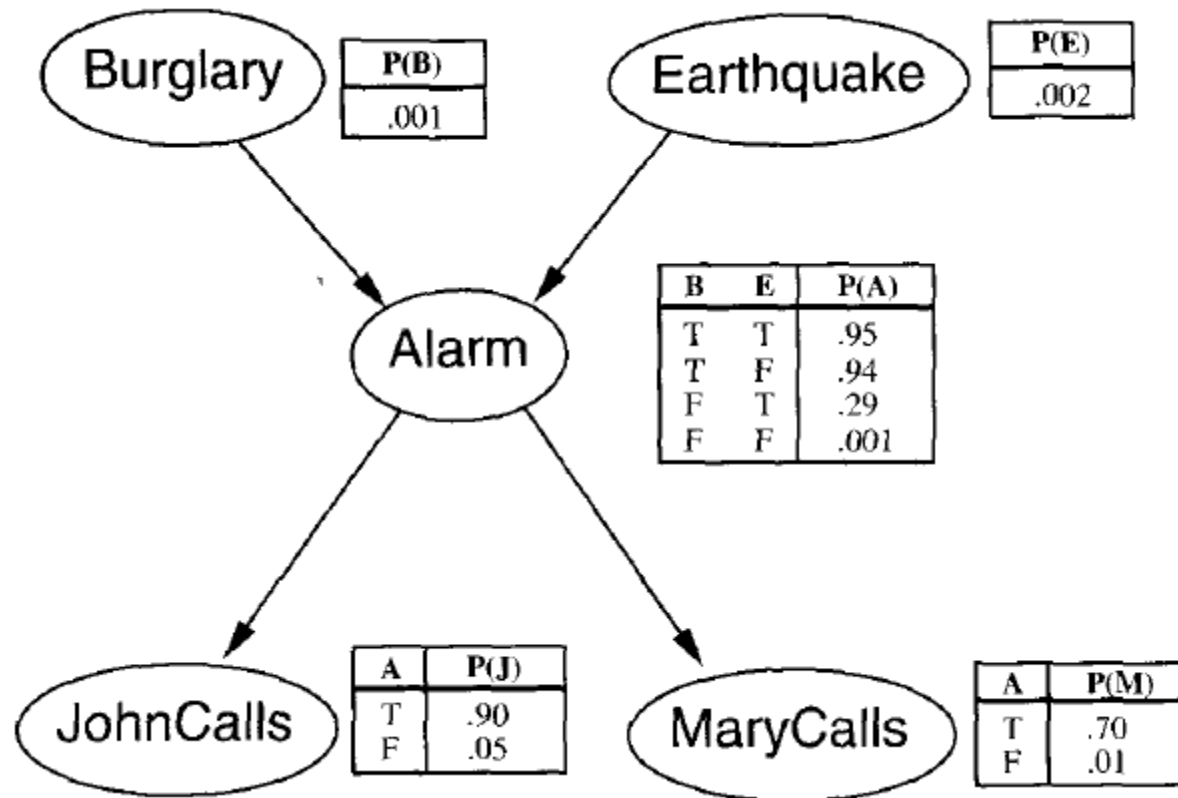
❖ Once we have specified the topology, we need to specify the conditional probability table or CPT for each node. Each row in the table contains the conditional probability of each node value for a conditioning case. A conditioning case is just a possible combination of values for the parent nodes (a miniature atomic event, if you like). For example, the conditional probability table for the random variable Alarm might look like this:

Burglary	Earthquake	$P(\text{Alarm} \text{Burglary}, \text{Earthquake})$	
		<i>True</i>	<i>False</i>
<i>True</i>	<i>True</i>	0.950	0.050
<i>True</i>	<i>False</i>	0.950	0.050
<i>False</i>	<i>True</i>	0.290	0.710
<i>False</i>	<i>False</i>	0.001	0.999

❖ (Each row in a conditional probability table must sum to 1, because the entries represent an exhaustive set of cases for the variable. Hence only one of the two numbers in each row shown above is independently specifiable. In general, a table for a Boolean variable with n Boolean parents contains 2^n independently specifiable probabilities. A

node with no parents has only one row, representing the prior probabilities of each possible value of the variable.)

Δίκτυα Πιθανοτήτων Bayes



Δίκτυα Πιθανοτήτων Bayes

❖ We can calculate the probability of the event that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Mary call

$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = P(J | A)P(M | A)P(A | \neg B \wedge \neg E)P(\neg B)P(\neg E)$$

$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 = 0.00062$$

Naive Bayes Classifiers

- Υπόθεση: το σύνολο εκπαίδευσης αποτελείται από στιγμιότυπα που περιγράφονται ως συζεύξεις των τιμών των χαρακτηριστικών, η ταξινόμηση βασίζεται σε ένα πεπερασμένο σύνολο κλάσεων V .
- Το έργο του ταξινομητή είναι να προβλέψει τη σωστή κλάση για ένα νέο στιγμιότυπο a_1, a_2, \dots, a_n .
- Βασικά ιδέα: εκχώρηση του στιγμιότυπου στην κλάση με την μεγαλύτερη πιθανότητα χρησιμοποιώντας των κανόνα Bayes.

$$\begin{aligned} v_{MAP} &= \arg \max_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n) \\ &= \arg \max_{v_j \in V} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)} \end{aligned}$$

Naive Bayes Classifiers

- Η εκτίμηση του $P(v_j)$ είναι απλή: υπολογίζουμε τη σχετική συχνότητα της κάθε κλάσης στόχου στο σύνολο εκπαίδευσης.
- Η εκτίμηση του $P(a_1, a_2, \dots, a_n | v_j)$ είναι δύσκολη: συνήθως δεν υπάρχουν αρκετά στιγμιότυπα για κάθε συνδυασμό των τιμών των χαρακτηριστικών στο σύνολο εκπαίδευσης.
- Υπόθεση Ανεξαρτησίας : οι τιμές των χαρακτηριστικών είναι υπό συνθήκη ανεξάρτητες δεδομένης της κλάσης στόχου : naive Bayes.

$$P(a_1, a_2, \dots, a_n | v_j)P(v_j) = \prod_i P(a_i | v_j)$$

- Ως εκ τούτου, παίρνουμε τον ακόλουθο ταξινομητή:

$$V_{nb} = \arg \max_{v_j \in V} P(v_j) \prod P(a_i | v_j)$$

Naive Bayes Classifiers

- Οι υπό συνθήκη πιθανότητες μπορούν να εκτιμηθούν άμεσα ως σχετικές συχνότητες:

$$P(a_i | v_j) = \frac{n_c}{n}$$

όπου n είναι ο συνολικός αριθμός των εμφανίσεων της κλάσης v_j στο σύνολο εκπαίδευσης, και n_c είναι ο αριθμός των στιγμιότυπων με χαρακτηριστικό a_i και κλάση v_j .

- *Πρόβλημα* : Κακή εκτίμηση εάν το n_c είναι πολύ μικρό.
- *Ακραία περίπτωση* : εάν $n_c = 0$, τότε η εκ των υστέρων πιθανότητα θα είναι 0.
- *Λύση* : χρήση της m -εκτίμησης των πιθανοτήτων:

$$P(a_i | v_j) = \frac{n_c + mp}{n + m}$$

p : εκ των προτέρων εκτίμηση της πιθανότητας

m : ισοδύναμο μέγεθος του δείγματος (σταθερό)

Ελλείψει άλλων πληροφοριών, υπόθεσε: $p = \frac{1}{k}$

όπου το k είναι ο αριθμός των τιμών που μπορεί να πάρει το χαρακτηριστικό a_i .

Naive Bayes Classifiers - Weather-Playing Game Example

- Τα χαρακτηριστικά είναι τα εξής : Outlook, Temperature, Humidity, Windy, και το υποκείμενο, Play, μπορεί να πάρει τιμές yes ή no.
- Το σύνολο δεδομένων που περιγράφει τις καιρικές συνθήκες και το αν κάποιος μπορεί να παίξει ένα υποθετικό παιχνίδι

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Naive Bayes Classifiers - Weather-Playing Game Example

Συχνότητες και Πιθανότητες για τα δεδομένα καιρού:

	outlook		temperature		humidity		windy		play				
	yes	no	yes	no	yes	no	yes	no	yes	no			
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- Με βάση τα δεδομένα εκπαίδευσης στον πίνακα, θέλουμε να ταξινομήσουμε το ακόλουθο στιγμιότυπο:

Outlook = sunny

Temperature = cool

Humidity = high

Windy = true

Naive Bayes Classifiers - Weather-Playing Game Example

Πρέπει να υπολογίσουμε το εξής :

$$V_{nb} = \arg \max_{u_j \in \{yes, no\}} P(u_j)P(x_1 = sunny | u_j)P(x_2 = cool | u_j)P(x_3 = high | u_j)P(x_4 = true | u_j)$$

Αρχικά υπολογίζουμε :

$$P(\text{play}=\text{yes})=9/14$$

$$P(\text{play}=\text{no})=5/14$$

$$P(x_1=\text{sunny}|\text{yes})=2/9$$

$$P(x_1=\text{sunny}|\text{no})=3/5$$

$$P(x_2=\text{cool}|\text{yes})=3/9$$

$$P(x_2=\text{cool}|\text{no})=1/5$$

$$P(x_3=\text{high}|\text{yes})=3/9$$

$$P(x_3=\text{high}|\text{no})=4/5$$

$$P(x_4=\text{true}|\text{yes})=3/9$$

$$P(x_4=\text{true}|\text{no})=3/5$$

	outlook		temperature		humidity		windy		play				
	yes	no	yes	no	yes	no	yes	no	yes	no			
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Naive Bayes Classifiers - Weather-Playing Game Example

Έπειτα υπολογίζουμε την καλύτερη κλάση:

$$P(\text{yes})P(\text{sunny}|\text{yes})P(\text{cool}|\text{yes})P(\text{high}|\text{yes})P(\text{true}|\text{yes}) \\ = 9/14 \cdot 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.0053$$

$$P(\text{no})P(\text{sunny}|\text{no})P(\text{cool}|\text{no})P(\text{high}|\text{no})P(\text{true}|\text{no}) \\ = 5/14 \cdot 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0206$$

Αφού $0.0206 > 0.0053$, το παράδειγμά μας ταξινομείται σαν 'NO'.

Expectation Maximization Algorithm

- Τα πιθανοτικά μοντέλα, όπως τα Μπεϋζιανά δίκτυα, συνήθως χρησιμοποιούνται για την μοντελοποίηση των δεδομένων.
- Μεγάλο μέρος της δημοτικότητας τους μπορεί να αποδοθεί στην ύπαρξη αποτελεσματικών και δυναμικών διαδικασιών για την εκμάθηση παραμέτρων από τις παρατηρήσεις.
- Συχνά, όμως, τα διαθέσιμα δεδομένα για την κατάρτιση ενός πιθανοτικού μοντέλου είναι ελλιπή.
- Ο αλγόριθμος expectation maximization δίνει τη δυνατότητα εκτίμησης των παραμέτρων στα πιθανοτικά μοντέλα με ελλιπή δεδομένα.

Expectation Maximization Algorithm

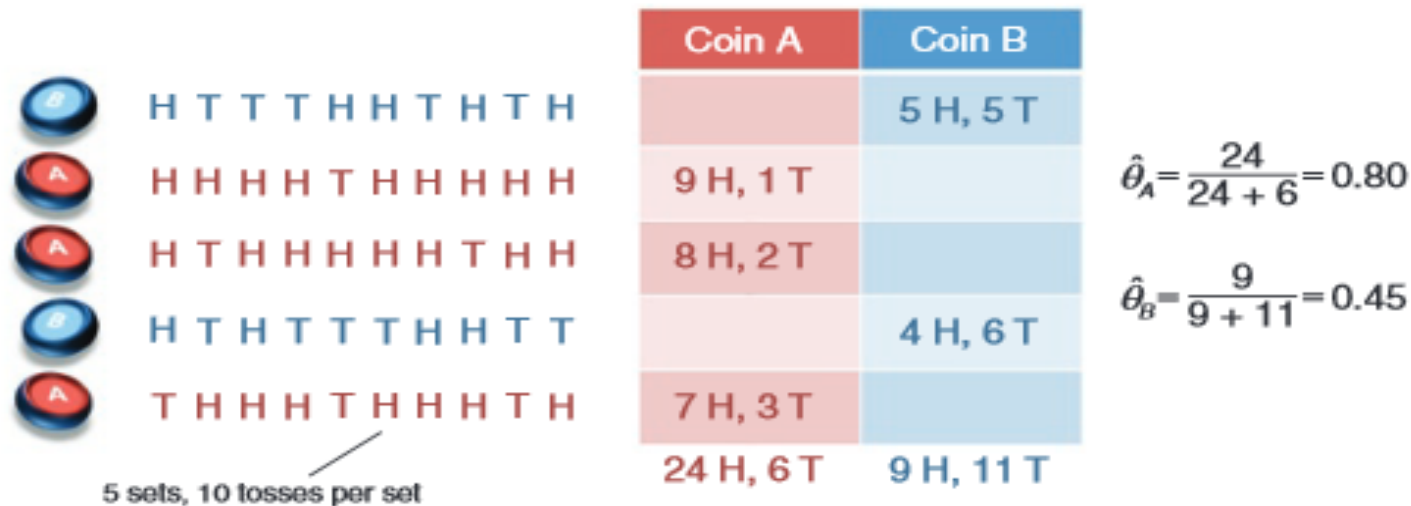
- Ο αλγόριθμος expectation maximization είναι μια φυσική γενίκευση της εκτίμησης μέγιστης πιθανοφάνειας για την περίπτωση ελλιπή δεδομένων.
 - Ειδικότερα, ο αλγόριθμος expectation maximization προσπαθεί να βρει τις παραμέτρους θ που μεγιστοποιούν την λογαριθμική πιθανότητα $\log P(x; \theta)$ των παρατηρούμενων δεδομένων.
- Σε γενικές γραμμές, το πρόβλημα βελτιστοποίησης στον αλγόριθμο expectation maximization είναι πιο δύσκολο από το πρόβλημα βελτιστοποίησης που χρησιμοποιείται στην εκτίμηση μέγιστης πιθανοφάνειας.
 - Σε περίπτωση με πλήρη δεδομένα, η αντικειμενική συνάρτηση $\log P(x, z; \theta)$ έχει ένα καθολικό μέγιστο, το οποίο μπορεί συχνά να βρεθεί σε κλειστή μορφή.
- Αντίθετα, στην περίπτωση ελλιπή δεδομένων η συνάρτηση $\log P(x; \theta)$ έχει πολλά τοπικά μέγιστα και καμία λύση σε κλειστή μορφή.
 - Για να το αντιμετωπίσει αυτό, ο αλγόριθμος expectation maximization μειώνει το δύσκολο έργο της βελτιστοποίησης του $\log P(x; \theta)$ σε μία ακολουθία απλούστερων υπό-προβλημάτων βελτιστοποίησης, οι αντικειμενικές συναρτήσεις των οποίων, έχουν μοναδικά καθολικά μέγιστα που μπορούν συχνά να υπολογιστούν σε κλειστή μορφή. Αυτά τα υπό-προβλήματα επιλέγονται κατά τρόπο που να εξασφαλίζονται οι αντίστοιχες λύσεις τους $\theta(1), \theta(2), \dots$ και να συγκλίνουν σε ένα τοπικό μέγιστο του $\log P(x; \theta)$.

Expectation Maximization Algorithm

- Πιο συγκεκριμένα, ο αλγόριθμος expectation maximization εναλλάσσεται ανάμεσα σε δύο φάσεις. Κατά τη διάρκεια του E-step, ο αλγόριθμος expectation maximization επιλέγει μια συνάρτηση g_t που χαμηλώνει τα όρια του $\log P(x; \theta)$ παντού, και για την οποία έχουμε $g_t(\theta(t)) = \log P(x; \theta(t))$.
 - Κατά τη διάρκεια του M-step, ο αλγόριθμος expectation maximization algorithm κινείται σε ένα νέο σύνολο παραμέτρων $\theta(t+1)$, που μεγιστοποιεί την g_t .
 - Δεδομένου ότι η τιμή του κάτω ορίου της g_t ταιριάζει με την αντικειμενική συνάρτηση στο $\theta(t)$, προκύπτει ότι $\log P(x; \theta(t)) = g_t(\theta(t)) \leq g_t(\theta(t+1)) = \log P(x; \theta(t+1))$ — έτσι η αντικειμενική συνάρτηση αυξάνει μονότονα κατά τη διάρκεια κάθε επανάληψης του αλγορίθμου expectation maximization!
- Συνοπτικά, ο αλγόριθμος expectation maximization εναλλάσσεται μεταξύ των βημάτων της πρόβλεψης μίας κατανομής πιθανοτήτων πάνω στο σύνολο των ελλιπή δεδομένων που δίνονται στο τρέχον μοντέλο (E-step) και στη συνέχεια στον εκ νέου υπολογισμό των παραμέτρων του μοντέλου με τη χρήση αυτών συνόλων (M-step).
 - Το όνομα 'E-step' προέρχεται από το γεγονός ότι κάποιος δεν χρειάζεται συνήθως να σχηματίζει την κατανομή πιθανότητας αποκλειστικά πάνω στα σύνολα, αλλά χρειάζεται μόνο να υπολογίσει τα «αναμενόμενα» στατιστικά στοιχεία πάνω σε αυτά τα σύνολα.
 - Ομοίως, η ονομασία 'M-step' προέρχεται από το γεγονός ότι το μοντέλο επανεκτίμησης μπορεί να θεωρηθεί ως η «μεγιστοποίηση» της αναμενόμενης λογαριθμικής-πιθανοφάνειας των δεδομένων.

EM - Coin-Flipping Experiment

- Σαν παράδειγμα, σκεφτείτε ένα απλό πείραμα ρίψης κέρματος στο οποίο μας δίνεται ένα ζευγάρι κέρματα A και B, τα οποία δεν είναι άρτια και εμφανίζουν μία τάση προς κεφαλή ή προς γράμματα.
- Εκτίμηση παραμέτρων για πλήρη δεδομένα.
- Η εκτίμηση μέγιστης πιθανοφάνειας. Για κάθε σύνολο δέκα ρίψεων, η διαδικασία μέγιστης πιθανοφάνειας συγκεντρώνει τις μετρήσεις των αποτελεσμάτων για τα κέρματα A και B ξεχωριστά. Αυτές οι μετρήσεις χρησιμοποιούνται στη συνέχεια για την εκτίμηση της τάσης των κερμάτων.



EM - Coin-Flipping Experiment

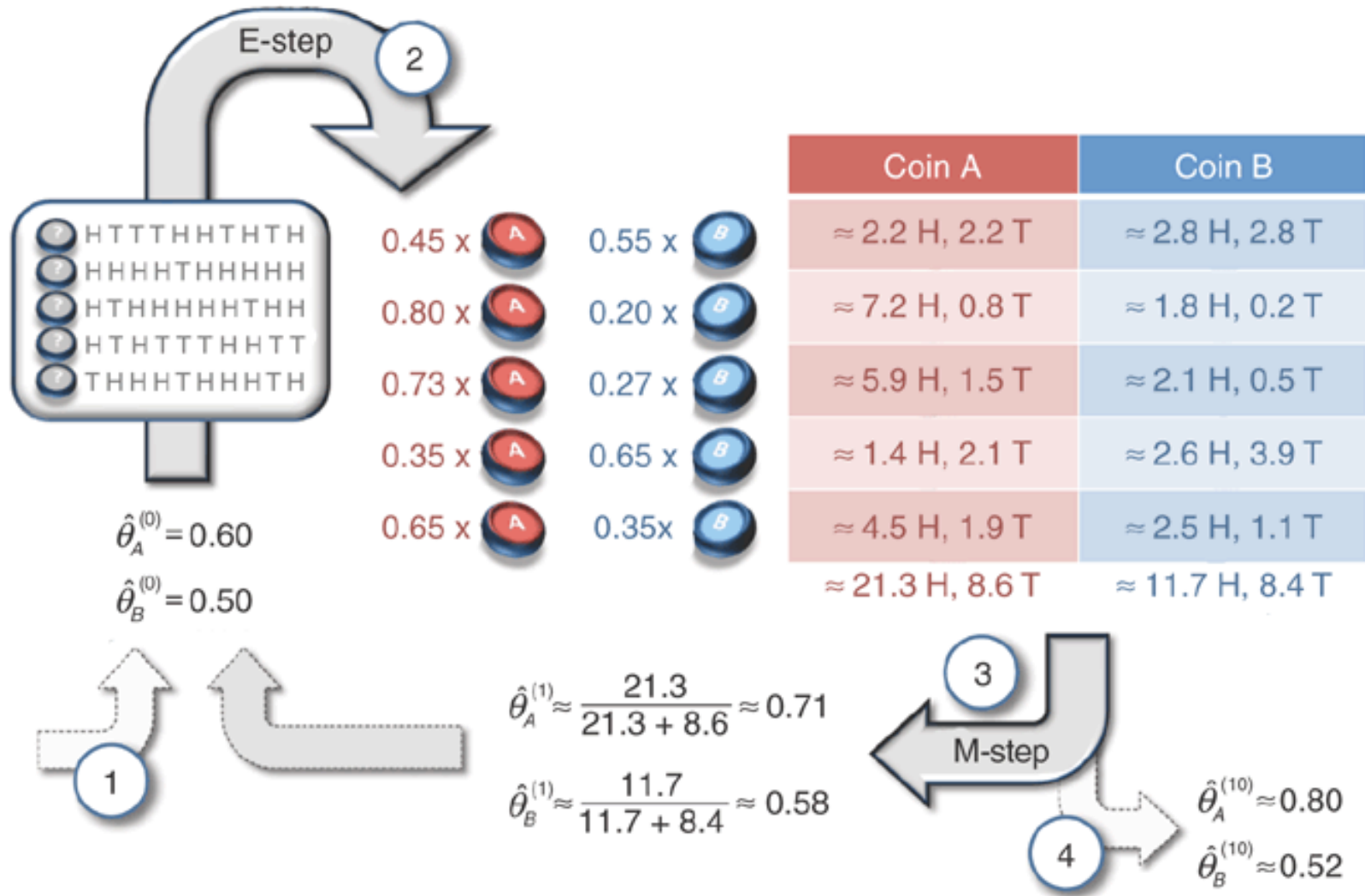
- Ας υποθέσουμε ένα πιο δύσκολο πρόβλημα.
 - Δεν ξέρουμε τις ταυτότητες των νομισμάτων που χρησιμοποιήθηκαν για κάθε σύνολο ρίψεων (τα αντιμετωπίζουμε ως κρυφές μεταβλητές)..

coin	flips	# coin A heads	# coin B heads
?	HTTTHHTHTH	?	?
?	HHHHTHHHHH	?	?
?	HTHHHHHTHH	?	?
?	HTHTTTHHTT	?	?
?	THHHTHHHTH	?	?

1. Αρχικά, ο EM ξεκινάει με μία τυχαία αρχική εκτίμηση των παραμέτρων.

$$\theta_A^{(0)} = 0.60, \theta_B^{(0)} = 0.50$$

EM - Coin-Flipping Experiment



EM - Coin-Flipping Experiment

2. Το E-step. Η κατανομή πιθανοτήτων πάνω στα δυνατά σύνολα υπολογίζεται με χρήση των τρεχόντων παραμέτρων. Οι μετρήσεις που παρουσιάζονται στον πίνακα είναι οι αναμενόμενοι αριθμοί για κορόνα ή γράμματα σύμφωνα με αυτή την κατανομή.

- Χρησιμοποιούμε τον τύπο:
$$p_i(k) = \binom{10}{k} \theta_i^k (1 - \theta_i)^{10-k}$$

- Ο διωνυμικός συντελεστής είναι ο ίδιος και για τα δύο κέρματα, έτσι απαλείφεται στην ομαλοποίηση, και μόνο ο λόγος από τους υπόλοιπους παράγοντες καθορίζει το αποτέλεσμα. Για παράδειγμα, στην τρίτη σειρά, έχουμε 8 κορόνες και 2 γράμματα. Δοθέντος των τρεχόντων παραμέτρων $\theta_A^{(0)} = 0.60$, $\theta_B^{(0)} = 0.50$, οι παράγοντες είναι:

$$p_A(8) = \theta_A^8 (1 - \theta_A)^{10-8} = 0.0026873856$$

$$p_B(8) = \theta_B^8 (1 - \theta_B)^{10-8} = 0.0009765625$$

- που μας δίνουν τα αποτελέσματα $\frac{p_A(8)}{p_A(8) + p_B(8)} = \frac{0.0026873856}{0.0026873856 + 0.0009765625} \approx 0.73$

- και $\frac{p_B(8)}{p_A(8) + p_B(8)} = \frac{0.0009765625}{0.0026873856 + 0.0009765625} \approx 0.27$

- για την τρίτη σειρά.

EM - Coin-Flipping Experiment

2. Πολλαπλασιάζουμε τις πιθανότητες που υπολογίσαμε στο βήμα 2 επί τον αριθμό των εμφανίσεων H (ή T).

flips	probability it was coin A	probability it was coin B	# heads attributed to A	# heads attributed to B
H T T T H H T H T H	0.45	0.55	2.2	2.8
H H H H T H H H H H	0.8	0.2	7.2	1.8
H T H H H H H T H H	0.73	0.27	5.9	2.1
H T H T T T H H T T	0.35	0.65	1.4	2.6
T H H H T H H H T H	0.65	0.35	4.5	2.5

EM - Coin-Flipping Experiment

4. Το M-step. Τώρα που έχουμε νέες εκτιμήσεις για τον αριθμό των Η και Τ για κάθε κέρμα μπορούμε να βελτιώσουμε τις εκτιμήσεις για τα Θ (δηλαδή τις πολώσεις των κερμάτων).

$$\theta_a^1 = \frac{2.2 + 7.2 + 5.9 + 1.4 + 4.5}{10 * (.45 + .8 + .73 + .35 + .65)} = 0.71$$

$$\theta_b^1 = \frac{2.8 + 1.8 + 2.1 + 2.6 + 2.5}{10 * (.55 + .2 + .27 + .65 + .35)} = 0.58$$

Με τις νέες τιμές Θ μπορούμε να επαναλάβουμε το E-step.

Μετά από 10 επαναλήψεις του E-step και του M-step, ο αλγόριθμος συγκλίνει.

$$\theta_A^{(10)} = 0.80, \theta_B^{(10)} = 0.52$$

Expectation Maximization Algorithm

- Ο αλγόριθμος expectation maximization εγγυάται μόνο την σύγκλιση σε ένα τοπικό μέγιστο της αντικειμενικής συνάρτησης (εκτός από εκφυλισμένες περιπτώσεις).
- Η εκτέλεση της διαδικασίας με τη χρήση πολλαπλών αρχικών παραμέτρων εκκίνησης είναι συχνά χρήσιμη, ομοίως η αρχικοποίηση των παραμέτρων με έναν τρόπο που να σπάει την συμμετρία σε μοντέλα είναι επίσης σημαντικός.
- Με αυτό το περιορισμένο σύνολο κινήσεων ο αλγόριθμος expectation maximization παρέχει ένα απλό και δυναμικό εργαλείο για την εκτίμηση των παραμέτρων στα μοντέλα με ελλιπή δεδομένα .
- Θεωρητικά, άλλες αριθμητικές τεχνικές βελτιστοποίησης, όπως η Newton- Raphson, θα μπορούσαν να χρησιμοποιηθούν αντί του αλγορίθμου expectation maximization. Στην πράξη, ωστόσο, ο αλγόριθμος expectation maximization έχει το πλεονέκτημα ότι είναι απλός, δυναμικός και εύκολος να υλοποιηθεί.

Συντελεστές Βεβαιότητας (*Certainty Factors*) (1/2)

- ❖ Αριθμητικές τιμές που εκφράζουν τη βεβαιότητα για την αλήθεια μιας πρότασης ή γεγονότος. Πρωτο-εισήχθησαν στο έμπειρο σύστημα MYCIN
- ❖ if γεγονός then υποθετικό συμπέρασμα με βεβαιότητα CF
Παράδειγμα: if πυρετός then γρίπη CF 0.8
- ❖ Παίρνουν τιμές στο διάστημα $[-1, +1]$
 - ❑ -1 : απόλυτη βεβαιότητα για το ψευδές της πρότασης.
 - ❑ $+1$: απόλυτη βεβαιότητα για την αλήθεια της πρότασης.
 - ❑ 0 : άγνοια.
- ❖ Τιμές βεβαιότητας και στην τιμή του γεγονότος του κανόνα:
Παράδειγμα: if πυρετός CF₁ 0.7 then γρίπη CF 0.8
 - ❑ τελική βεβαιότητα κανόνα: $0.7 \times 0.8 = 0.56$
- ❖ Αν υπάρχουν περισσότερα από ένα γεγονότα στο αριστερό τμήμα του κανόνα τα οποία συνδέονται με AND (ή OR) τότε ως συντελεστής βεβαιότητας του αριστερού τμήματος θεωρείται η μικρότερη (ή η μεγαλύτερη) τιμή CF που εμφανίζεται.

Συντελεστές Βεβαιότητας (*Certainty Factors*) (2/2)

❖ Αν δύο διαφορετικοί κανόνες συνάγουν το ίδιο υποθετικό συμπέρασμα με βεβαιότητες CF_p και CF_n , τότε η συνολική βεβαιότητα είναι:

- ❑ Αν CF_p και $CF_n > 0$, τότε: $CF = CF_p + CF_n(1 - CF_p) = CF_p + CF_n - CF_n CF_p$
- ❑ Αν CF_p και $CF_n < 0$, τότε: $CF = CF_p + CF_n(1 + CF_p) = CF_p + CF_n + CF_n CF_p$
- ❑ Αν CF_p και CF_n ετερόσημα, τότε:
$$CF = \frac{CF_p + CF_n}{1 - \min(|CF_p|, |CF_n|)}$$

❖ Παράδειγμα: if πυρετός then γρίπη CF 0.8
if βήχας then γρίπη CF 0.5

❖ **Συμπερασματικά:**

- ❑ Αντί για συχνότητες εμφάνισης γεγονότων που πρέπει να μετρηθούν, χρησιμοποιούνται συντελεστές βεβαιότητας που έχουν εκτιμηθεί από ειδικούς.
- ❑ Οι υπολογισμοί κατά το συνδυασμό βεβαιοτήτων είναι απλούστεροι, λόγω της παραδοχής της ανεξαρτησίας των γεγονότων.
- ❑ Πρέπει να αποφεύγεται η ταυτόχρονη χρήση κανόνων που αναστρέφουν τη σχέση αιτίας-αποτελέσματος. Π.χ. *if A then B* και *if B then A*

Παράδειγμα

❖ Έστω ότι δύο κανόνες οδηγούν στο ίδιο υποθετικό συμπέρασμα B, κάτω όμως από διαφορετικές παραδοχές, δηλαδή:

if A then B CF 0.8

if C AND D AND E then B CF 0.6

❖ Αν ο χρήστης εισάγει τα δεδομένα A, C, D και E με βεβαιότητες:
 $CF(A)=0.5$, $CF(C)=0.9$, $CF(D)=0.7$ και $CF(E)=0.5$ τότε:

❖ Η ενεργοποίηση του πρώτου κανόνα δίνει: $CF_p(B)=0.5 * 0.8 = 0.4$

❖ Η ενεργοποίηση του δεύτερου κανόνα δίνει:
 $CF_n(B)=0.6 * \min(0.9, 0.7, 0.5) = 0.6 * 0.5 = 0.3$

❖ Επειδή τα CF_p και CF_n είναι και τα δύο θετικά, η συνολική βεβαιότητα του υποθετικού συμπεράσματος B θα είναι:

$$CF(B) = 0.4 + 0.3 - (0.4 \times 0.3) = 0.58$$

Προσέγγιση Dempster-Shafer (D-S) (1/2)

- ❖ Δεν απαιτείται η συλλογή όλων των απλών και των υπό συνθήκη πιθανοτήτων.
- ❖ Λογισμός με αριθμητικές τιμές πεποίθησης (*belief*)
- ❖ Πλαίσιο διάκρισης (*frame of discernment*)
- ❖ Αν $U = \{A, B, C\}$ πιθανές ασθένειες τότε Pow είναι το σύνολο των υποσυνόλων του U :

$$\text{Pow}(U) = \{ \{\}, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\} \}$$

πιθανές διαγνώσεις.

- ❖ Διαζευγμένες Προτάσεις: $\{A, B\}$ σημαίνει "ασθένεια A ή B".
- ❖ Στοιχεία του U που δεν ανήκουν σε ένα στοιχείο του $\text{Pow}(U)$, (π.χ. η ασθένεια C στο $\{A, B\}$), κάνουν σαφή την άρνηση του αντίστοιχου υποθετικού συμπεράσματος.
- ❖ $\{\}$: *null hypothesis*

Προσέγγιση Dempster-Shafer (2/2)

- ❖ Η *βασική κατανομή πιθανότητας* (*basic probability assignment - bpa*) είναι μία απεικόνιση: $m: Pow(U) \rightarrow [0,1]$ η οποία αναθέτει μία τιμή πεποίθησης για κάθε στοιχείο του $Pow(U)$
- ❖ Είναι δηλαδή το μέτρο της πεποίθησης που υπάρχει για το κατά πόσο ισχύει το υποθετικό συμπέρασμα που εκφράζεται με το συγκεκριμένο στοιχείο του U .
 - ❑ η πεποίθηση $m(\{A, B\})=0.3$, δε μοιράζεται στα $\{A\}$ και $\{B\}$ αλλά αφορά το $\{A, B\}$.
 - ❑ ισχύει $m(\{\})=0$
 - ❑
$$\sum_{X \in Pow(U)} m(X) = 1$$
 - ❑ Επειδή θεωρούμε ότι το υποθετικό συμπέρασμα βρίσκεται κάπου μέσα στα στοιχεία του $Pow(U)$
- ❖ Η *συνολική πεποίθηση* (*belief*) ότι ένα στοιχείο του U ανήκει στο X καθώς και στα τυχόν υποσύνολα του X συμβολίζεται με $Bel(X)$

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

Dempster-Shafer vs. Bayes

- ❖ Bayes: η απουσία άλλων ενδείξεων για τις δυνατές εκδοχές τις καθιστά ισοπίθανες.
- ❖ Dempster-Shafer: η απουσία κάποιων ενδείξεων θέτει την πιθανότητα (*likelihood*) κάθε εκδοχής κάπου στο διάστημα $[0, 1]$.

Κανόνας Dempster-Shafer

- ❖ Αν m_1 και m_2 δύο ανεξάρτητες εκτιμήσεις (βασικές κατανομές πιθανότητας) που αποδίδουν κάποιο βαθμό πεποίθησης στα στοιχεία του $\text{Pow}(U)$, τότε αυτές συνδυάζονται σε μία τρίτη εκτίμηση $m_3 = m_1 \oplus m_2$ με τρόπο που ορίζεται με τον κανόνα *D-S*:

$$m_3(A) = m_1 \oplus m_2(A) = \frac{\sum_{X, Y \in U: X \cap Y = A} m_1(X) \cdot m_2(Y)}{1 - \sum_{X, Y \in U: X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}$$

Παράδειγμα: Διάγνωση Ασθένειας

- ❖ Έστω $U = \{A, B, C\}$ το σύνολο των δυνατών ασθενειών που μπορεί να διαγνωσθούν.
- ❖ Πιθανές Διαγνώσεις $Pow(U) = \{\{\}, \{A\}, \{B\}, \{C\}, \{A,B\}, \{A,C\}, \{B,C\}, \{A,B,C\}\}$
- ❖ $m(\{\{\}, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}) = 1$
 - υποδηλώνει τη βεβαιότητα ότι η διάγνωση βρίσκεται κάπου στα στοιχεία του $Pow(U)$ αλλά ελλείπει άλλων ενδείξεων δεν είναι δυνατό να δοθεί ιδιαίτερη βαρύτητα σε κάποιο
 - Bayes: θα έπρεπε κάθε στοιχείο του $Pow(U)$ να θεωρηθεί ισοπίθανο
- ❖ Έστω ότι γίνεται διαθέσιμη επιπλέον πληροφορία, (π.χ. πραγματοποιούνται ιατρικές εξετάσεις) και προκύπτει ότι η ασθένεια είναι μία από τις A ή B με βαθμό πίστης 0.7
 - $m1(\{A, B\}) = 0.7$
 - $m1(\{\{\}, \{A\}, \{B\}, \{C\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}) = 0.3$
 - Δηλαδή, η έλλειψη πίστης σε ένα από τα υποθετικά συμπεράσματα του $Pow(U)$, ισοδυναμεί αυτόματα με ισόποσο βαθμό πίστης στα υπόλοιπα στοιχεία του $Pow(U)$, χωρίς όμως να δίνεται ιδιαίτερη προτίμηση σε κάποιο από αυτά.
 - Bayes: απαιτείται ο υπολογισμός μεγάλου αριθμού υπό συνθήκη πιθανοτήτων, κάτι που είναι υπολογιστικά ακριβό και πολλές φορές αδύνατο.
- ❖ Πώς μπορεί να συνδυαστούν δύο ανεξάρτητες εκτιμήσεις (π.χ. δύο ιατρών) σε μία;

Παράδειγμα: Συνδυασμός Διαγνώσεων

❖ Έστω ότι δύο γιατροί εξετάζουν ανεξάρτητα τον ασθενή και δίνουν την εκτίμησή τους m_1 και m_2 αντίστοιχα, για την αρρώστια από την οποία αυτός πάσχει.

Δυνατές περιπτώσεις διάγνωσης	Γιατρός 1		Γιατρός 2	
	m_1	Bel_1	m_2	Bel_2
{A}	0.05	0.05	0.15	0.15
{B}	0	0	0	0
{C}	0.05	0.05	0.05	0.05
{A, B}	0.15	0.2	0.05	0.2
{A, C}	0.1	0.2	0.2	0.4
{B, C}	0.05	0.1	0.05	0.1
{A, B, C}	0.6	1	0.5	1

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

- ❖ π.χ. $Bel_1(\{A,B\}) = m_1(\{A,B\}) + m_1(\{A\}) + m_1(\{B\}) = 0.015 + 0.05 + 0 = 0.2$
- ❖ Οι δύο ανεξάρτητες εκτιμήσεις m_1 και m_2 μπορεί να συνδυαστούν στην m_3 χρησιμοποιώντας τον κανόνα Dempster-Shafer:

$$m_3(A) = m_1 \oplus m_2(A) = \frac{\sum_{X,Y \in Pow(U): X \cap Y = A} m_1(X) \cdot m_2(Y)}{1 - \sum_{X,Y \in Pow(U): X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}$$

Παράδειγμα: Συνδυασμός Εκτιμήσεων (1/2)

$m_3 = m_1 \oplus m_2$		m_1						
		{A}	{B}	{C}	{A,B}	{A,C}	{B,C}	{A,B,C}
m_2		0.05	0	0.05	0.15	0.1	0.05	0.6
{A}	0.15	{A} .0075	{ } 0	{ } .0075	{A} .0225	{A} .015	{ } .0075	{A} .09
{B}	0	{ } .0	{B} 0	{ } .0	{B} .0	{ } .0	{B} .0	{B} .0
{C}	0.05	{ } .0025	{ } 0	{C} .0025	{ } .0075	{C} .005	{C} .0025	{C} .03
{A,B}	0.05	{A} .0025	{B} 0	{ } .0025	{A,B} .0075	{A} .005	{B} .0025	{A,B} .03
{A,C}	0.2	{A} .01	{ } 0	{C} .01	{A} .03	{A,C} .02	{C} .01	{A,C} .012
{B,C}	0.05	{ } .0025	{B} 0	{C} .0025	{B} .0075	{C} .005	{B,C} .0025	{B,C} .03

{A,B,C}	0.5	{A} .025	{B} 0	{C} .025	{A,B} .075	{A,C} .05	{B,C} .025	{A,B,C} .3
---------	-----	----------	-------	----------	------------	-----------	------------	------------

Παράδειγμα: Συνδυασμός Εκτιμήσεων (2/2)

Δυνατές περιπτώσεις διάγνωσης	m_3	Bel_3
{A}	0.21	0.21
{B}	0.01	0.01
{C}	0.09	0.09
{A, B}	0.12	0.34
{A, C}	0.20	0.50
{B, C}	0.06	0.16
{A, B, C}	0.31	1.00

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

- ❖ Η αρχική εκτίμηση ότι η ασθένεια είναι μία από τις A ή B αποδυναμώθηκε.
 - η διάγνωση βρίσκεται μάλλον στο σύνολο $\{A, C\}$
 - επειδή $Bel_3(\{A\}) > Bel_3(\{C\})$, αρχίζει να διαφαίνεται ότι η τελική διάγνωση είναι η A

- ❖ Η παραπάνω συνδυασμένη εκτίμηση μπορεί να συνδυαστεί εκ νέου με μια άλλη εκτίμηση (π.χ. 3^{ου} ιατρού).

Ασάφεια (*Fuzziness*)

- ❖ Έννοια που σχετίζεται με την ποσοτικοποίηση της πληροφορίας και οφείλεται κυρίως σε μη-ακριβή (*imprecise*) δεδομένα.
 - "Ο Νίκος είναι ψηλός"
- ❖ Το πρόβλημα οφείλεται στην αντίληψη που έχει ο καθένας για λεκτικούς προσδιορισμούς ποσοτικών μεγεθών (*σημασιολογική ασάφεια*)
- ❖ Εγγενές χαρακτηριστικό της γλώσσας.
- ❖ *Ασαφής Λογική (fuzzy logic)*: υπερέννολο της κλασικής λογικής
 - χειρίζεται τιμές αληθείας μεταξύ του "απολύτως αληθούς" και του "απολύτως ψευδούς".
- ❖ *Θεωρία Ασαφών Συνόλων (Fuzzy Set Theory)* - Lofti Zadeh '60

Βασικές Έννοιες Ασαφών Συνόλων

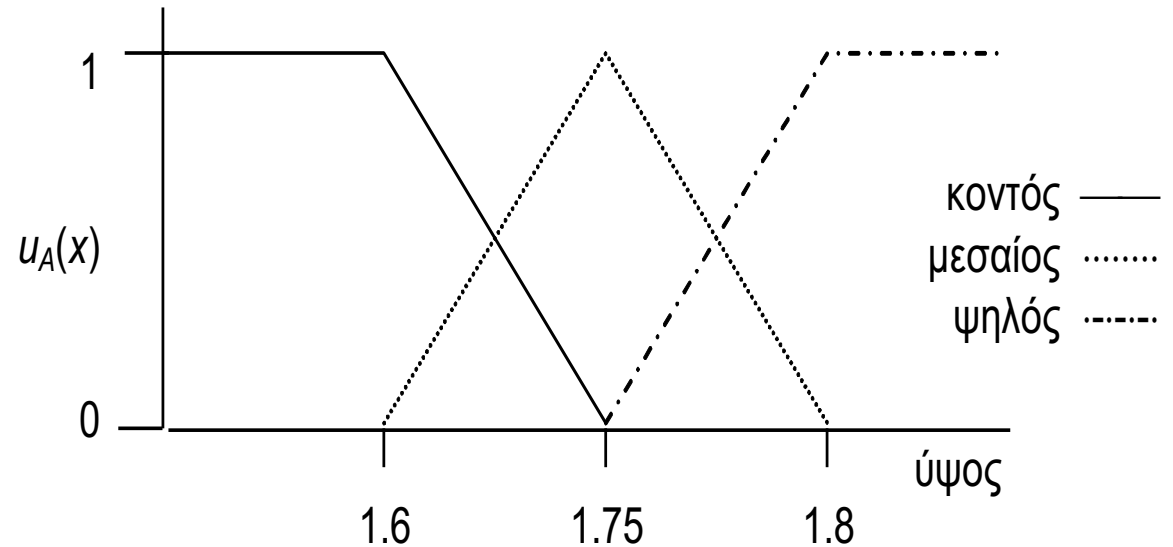
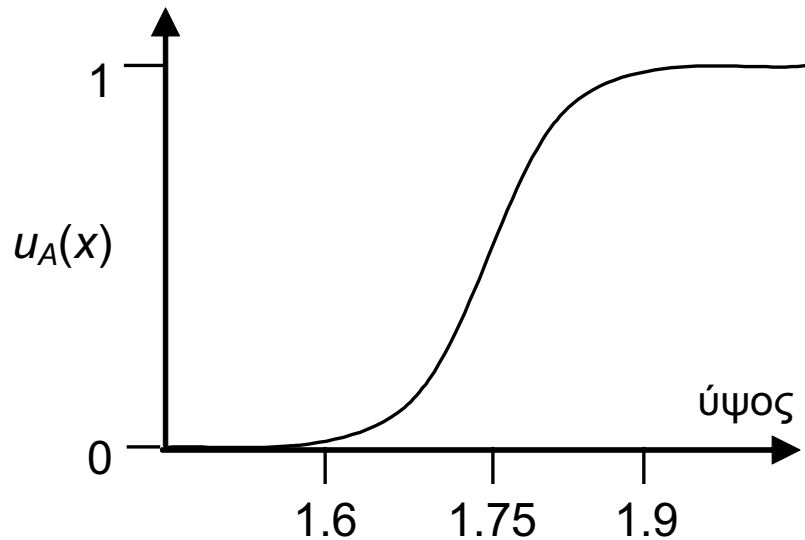
❖ *Ασαφές Σύνολο (fuzzy set) A*: ένα σύνολο διατεταγμένων ζευγών $(x, u_A(x))$ όπου $x \in X$ και $u_A(x) \in [0, 1]$.

- ❑ Το σύνολο X περιλαμβάνει όλα τα αντικείμενα στα οποία μπορεί να γίνει αναφορά.
- ❑ $u_A(x)$: *βαθμός αληθείας (degree of truth)* - τιμές στο διάστημα $[0, 1]$.
- ❑ Η συνάρτηση u_A ονομάζεται *συνάρτηση συγγένειας (membership function)*.

❖ Προέλευση u_A :

- ❑ Υποκειμενικές εκτιμήσεις
- ❑ Προκαθορισμένες (ad hoc) μορφές
- ❑ Συχνότητες εμφανίσεων και πιθανότητες
- ❑ Φυσικές μετρήσεις
- ❑ Διαδικασίες μάθησης και προσαρμογής (νευρωνικά δίκτυα)

Αναπαράσταση Ασαφών Συνόλων



- ❖ Αναλυτική έκφρασης της u_A
- ❖ Απλούστευση: τμηματικώς γραμμικής απεικόνιση της u_A
- ❖ Σύνολο ζευγών της μορφής $u_A(x)/x$
 - Π.χ. ψηλός = $\{0/1.7, 0/1.75, 0.33/1.8, 0.66/1.85, 1/1.9, 1/1.95\}$
- ❖ Με ζεύγη της μορφής $(x, u_A(x))$:
 - Π.χ. ψηλός = $\{ (1.7, 0), (1.75, 0), (1.8, 0.33), (1.85, 0.66), (1.9, 1), (1.95, 1) \}$

Ασαφείς Σχέσεις (1/2)

- ❖ Ασαφή σύνολα ορισμένα σε πεδία αναφοράς ανώτερης διάστασης.
- **Παράδειγμα:** $R = "x \text{ είναι βαρύτερο από } y"$ $x \in X, y \in Y$ και $R \in X \cdot Y$
- ❖ Αναπαράσταση της R , σε μορφή πίνακα:

$$R = \begin{bmatrix} u_R(x_1, y_1) & u_R(x_1, y_2) & \cdots & u_R(x_1, y_n) \\ u_R(x_2, y_1) & u_R(x_2, y_2) & \cdots & u_R(x_2, y_n) \\ \vdots & \vdots & \vdots & \vdots \\ u_R(x_m, y_1) & u_R(x_m, y_2) & \cdots & u_R(x_m, y_n) \end{bmatrix}$$

- ❖ **Σύνθεση (composition) Ασαφών Σχέσεων:** συνδυασμός ασαφών σχέσεων.
- Πρέπει να προσδιοριστεί η συνάρτηση συγγένειας $u_R(x,z)$ της R , με χρήση των συναρτήσεων συγγένειας των R_1 και R_2 , δηλαδή των $u_{R_1}(x,y)$ και $u_{R_2}(y,z)$.

Ασαφείς Σχέσεις (2/2)

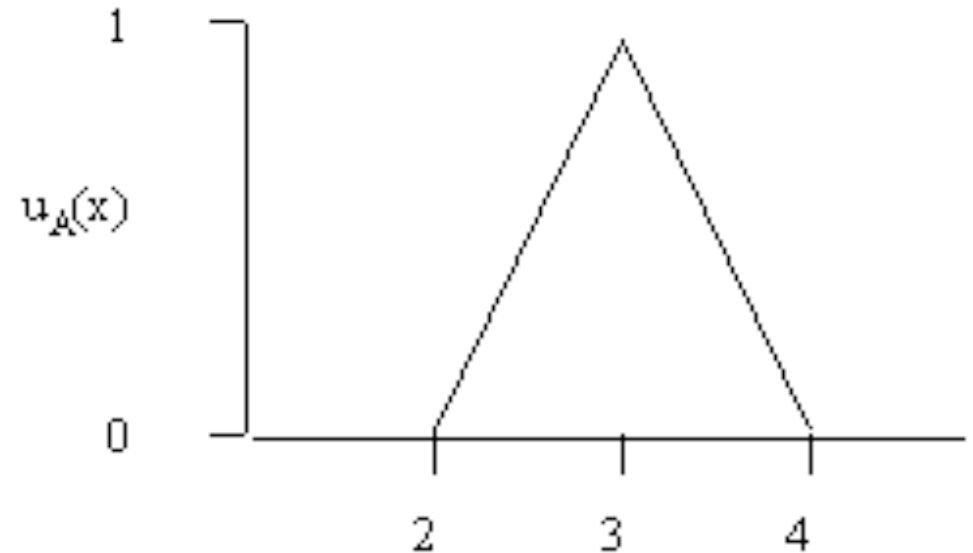
- ❖ Σύνθεση *max-min* (*max-min composition*)
- ❖ Σύνθεση *max-product* (*max-product composition*).
- ❖ Αν $R_1(x,y)$ και $R_2(y,z)$ είναι δύο ασαφείς σχέσεις ορισμένες στα σύνολα $X \cdot Y$ και $Y \cdot Z$ αντίστοιχα, τότε η σύνθεσή τους δίνει μία νέα σχέση $R_1 \circ R_2$ ορισμένη στο $X \cdot Z$ με συνάρτηση συγγένειας:

□ Σύνθεση *max-min*:
$$u_{R_1 \circ R_2}(x, z) = \bigvee_y [u_{R_1}(x, y) \wedge u_{R_2}(x, y)]$$

□ Σύνθεση *max-product*:
$$u_{R_1 \circ R_2}(x, z) = \bigvee_y [u_{R_1}(x, y) \cdot u_{R_2}(x, y)]$$

Ασαφείς Μεταβλητές και Ασαφείς Αριθμοί

- ❖ *Ασαφής Μεταβλητή (fuzzy variable)*: οι τιμές τις ορίζονται με ασαφή σύνολα.
 - Π.χ. τα ασαφή σύνολα {κοντός, μεσαίος, ψηλός} θα μπορούσαν να είναι το πεδίο τιμών της ασαφούς μεταβλητής "ύψος".
 - "ύψος": λεκτική (*linguistic*) μεταβλητή.
- ❖ *Ασαφείς αριθμοί (fuzzy numbers)*: ασαφή υποσύνολα του συνόλου των πραγματικών αριθμών. Π.χ. "Ασαφές 3" στο σχήμα.
- ❖ μη ασαφείς τιμές: *crisp* (σαφείς, συγκεκριμένες).



Η Αρχή της Επέκτασης (1/2)

❖ Επέκταση των εννοιών και των υπολογιστικών τεχνικών των κλασικών μαθηματικών στο πλαίσιο των ασαφών.

❖ συνάρτηση f : ορίζει απεικόνιση του $X = \{x_1, x_2, \dots, x_n\}$ στο $Y = \{y_1, y_2, \dots, y_n\}$, έτσι ώστε $y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$.

❖ ασαφές σύνολο A ορισμένο στα στοιχεία του X

$$A = \{ u_A(x_1)/x_1, u_A(x_2)/x_2, \dots, u_A(x_n)/x_n \}$$

❖ Αν η είσοδος x της συνάρτησης f γίνει ασαφής μέσω του συνόλου A , τότε τι συμβαίνει με την έξοδο y ;

❖ **Αρχή Επέκτασης:** υπολογισμός ασαφούς συνόλου B με εφαρμογή της f στο A .

$$B = f(A) = \{ u_A(x_1)/f(x_1), u_A(x_2)/f(x_2), \dots, u_A(x_n)/f(x_n) \}$$

□ δηλαδή, κάθε $y_i = f(x_i)$ γίνεται ασαφές σε βαθμό $u_A(x_i)$

□ πρακτικά, η $u_B(y)$ προκύπτει από την $u_A(x)$ όπου το x αντικαθίσταται με την έκφραση που προκύπτει για αυτό από την επίλυση της f ως προς x

Η Αρχή της Επέκτασης (2/2)

❖ Ειδικές Περιπτώσεις

□ αν περισσότερα του ενός διαφορετικά x (έστω τα x_m και x_n) δίνουν μέσω της συνάρτησης f το ίδιο y (έστω το y_0), τότε: $u_B(y_0) = u_A(x_m) \vee u_A(x_n)$.

• η μέγιστη τιμή συγγένειας των x_m και x_n στο A επιλέγεται ως βαθμός συγγένειας του y_0 στο B

□ αν για κάποιο y_0 του B δεν υπάρχει x_0 του A τέτοιο ώστε $y_0 = f(x_0)$, τότε η τιμή συγγένειας του B στο y_0 είναι μηδέν.

❖ Γενίκευση σε περισσότερες διαστάσεις:

□ αν υπάρχουν οι μεταβλητές u, v, \dots, w ορισμένες στα σύνολα U, V, \dots, W αντίστοιχα

□ m διαφορετικά ασαφή σύνολα A_1, A_2, \dots, A_m ορισμένα στο $U \cdot V \cdot \dots \cdot W$

□ η πολυπαραμετρική συνάρτηση $y = f(u, v, \dots, w)$

τότε η ασαφοποίηση του χώρου των y , δηλαδή η συνάρτηση συγγένειας του συνόλου B , ορίζεται ως εξής:

$$u_B(y) = \bigvee_{U \times V \times \dots \times W} [u_{A_1}(u) \wedge u_{A_2}(v) \wedge \dots \wedge u_{A_m}(w)] / f(u, v, \dots, w)$$

Παράδειγμα Χρήσης Αρχής Επέκτασης (1/2)

❖ Πρόσθεση των αριθμών A:"ασαφές 3" και B:"ασαφές 7"

□ A: "ασαφές 3" = { 0/1, 0.5/2, 1/3, 0.5/4, 0/5 }

□ B: "ασαφές 7" = { 0/5, 0.5/6, 1/7, 0.5/8, 0/9 }

❖ Κατασκευάζεται ο πίνακας:

	B	A	x=1	x=2	x=3	x=4	x=5
y=5	0	0	0 ⁰	0 ^{0.5}	0 ¹	0 ^{0.5}	0 ⁰
y=6	0.5	0	0.5 ⁰	0.5 ^{0.5}	0.5 ¹	0.5 ^{0.5}	0.5 ⁰
y=7	1	0	1 ⁰	1 ^{0.5}	1 ¹	1 ^{0.5}	1 ⁰
y=8	0.5	0	0.5 ⁰	0.5 ^{0.5}	0.5 ¹	0.5 ^{0.5}	0.5 ⁰
y=9	0	0	0 ⁰	0 ^{0.5}	0 ¹	0 ^{0.5}	0 ⁰

❖ Σύμφωνα με την αρχή της επέκτασης θα είναι:

$$u_{C=A+B}(z) = \bigvee_{z=x+y} [u_A(x) \wedge u_B(y)] / (x+y)$$

Παράδειγμα Χρήσης Αρχής Επέκτασης (2/2)

❖ Έστω $z=9$. Υπάρχουν δύο συνδυασμοί x και y που μας δίνουν άθροισμα 9.

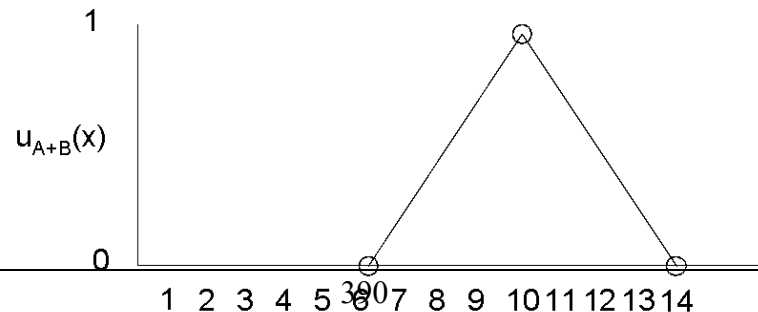
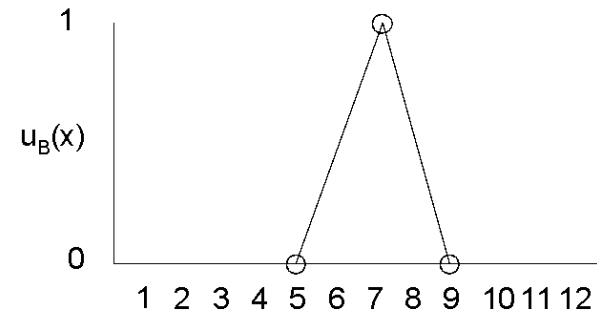
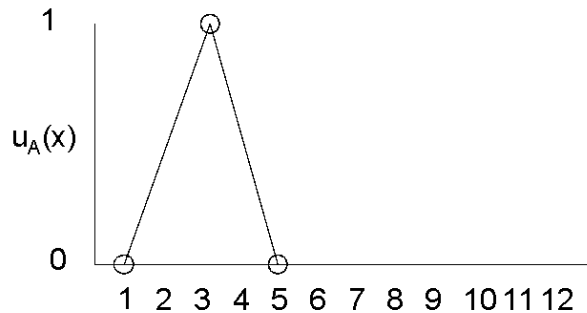
❖ αρχή της επέκτασης:

$$\begin{aligned} u_{A+B}(9) &= \max(\min(u_A(4), u_B(5)), \min(u_A(3), u_B(6)), \min(u_A(2), u_B(7)), \min(u_A(1), u_B(8))) = \\ &= \max(\min(0.5, 0), \min(1, 0.5), \min(0.5, 1), \min(0, 0.5)) = \\ &= \max(0, 0.5, 0.5, 0) = 0.5 \end{aligned}$$

□ Άρα: ο βαθμός συγγένειας του $z=9$ στο ασαφές σύνολο $C=A+B$ είναι 0.5.

❖ όμοια για τα υπόλοιπα στοιχεία του πίνακα, οπότε προκύπτει:

$$u_{A+B}(z) = \{ 0/8, 0.5/9, 1/10, 0.5/11, 0/12 \}$$



Παράδειγμα Άμεσης Εφαρμογής Αρχής Επέκτασης

❖ Έστω η $y=f(x)$:

$$y = f(x) = \sqrt{1 - \frac{x^2}{4}}$$

❖ Έστω ότι το x γίνεται ασαφές μέσω της:

$$u_A(x) = 1/2 \cdot x$$

❖ Η έξοδος y της f γίνεται ασαφής με το $u_B(y)$ να προκύπτει με επίλυση της f ως προς x :

$$x = 2\sqrt{1 - y^2}$$

και αντικατάσταση στην $u_A(x)$:

$$u_B(y) = u_A(2\sqrt{1 - y^2}) = \frac{1}{2} \cdot 2\sqrt{1 - y^2} = \sqrt{1 - y^2}$$

Ασαφείς Προτάσεις και Ασαφείς Κανόνες

- ❖ *Ασαφής πρόταση* είναι αυτή που θέτει μια τιμή σε μια ασαφή μεταβλητή.
- ❖ *Ασαφής κανόνας (fuzzy rule)*: είναι μία υπό συνθήκη έκφραση που συσχετίζει δύο ή περισσότερες ασαφείς προτάσεις.
 - "Εάν η ταχύτητα είναι μέτρια τότε η πίεση στα φρένα να είναι μέτρια"
- ❖ Η αναλυτική περιγραφή ενός ασαφούς κανόνα if-then είναι μία ασαφής σχέση $R(x,y)$ που ονομάζεται *σχέση συνεπαγωγής (implication relation)*.
 - ❖ Γενική της μορφή της σχέσης (συνάρτησης) συνεπαγωγής:
$$R(x,y) \equiv u(x,y) = \varphi(u_A(x), u_B(y))$$
 - φ : *τελεστής συνεπαγωγής (implication operator)*

Μερικοί Ασαφείς Τελεστές Συνεπαγωγής

Όνομασία Τελεστή	Αναλυτική Έκφραση του $\varphi[u_A(x), u_B(y)]$
φ_m : Zadeh Max-Min	$(u_A(x) \wedge u_B(y)) \vee (1 - u_A(x))$
φ_c : Mandani Min	$u_A(x) \wedge u_B(y)$
φ_p : Larsen Product	$u_A(x) \cdot u_B(y)$
φ_a : Arithmetic	$1 \wedge (1 - u_A(x) + u_B(y))$
φ_b : Boolean	$(1 - u_A(x)) \vee u_B(y)$

Συλλογιστικές Διαδικασίας GMP και GMT

❖ Γενική μορφή προβλημάτων κατά τη συλλογιστική με ασαφείς κανόνες:

□ if x is A then y is B

□ x is A' y is B' (?)

μέσω της συλλογιστικής διαδικασίας GMP (Generalized Modus Ponens - GMP): $B' = A' \circ R(x, y)$

□ if x is A then y is B

□ x is A' (?) y is B'

μέσω της συλλογιστικής διαδικασίας GMT (Generalized Modus Tollens - GMT): $A' = R(x, y) \circ B'$

❖ Η σχέση συνεπαγωγής $R(x,y)$ που έχει επιλεγεί να χρησιμοποιηθεί, πρέπει να συνδυαστεί (σύνθεση) με την κατά περίπτωση γνωστή παράμετρο (A' ή B') ώστε να υπολογιστεί η άγνωστη παράμετρος.

Σύνοψη Ασαφούς Συλλογιστικής Διαδικασίας

❖ Με βάση έναν ασαφή κανόνα της μορφής:

"if x is A then y is B "

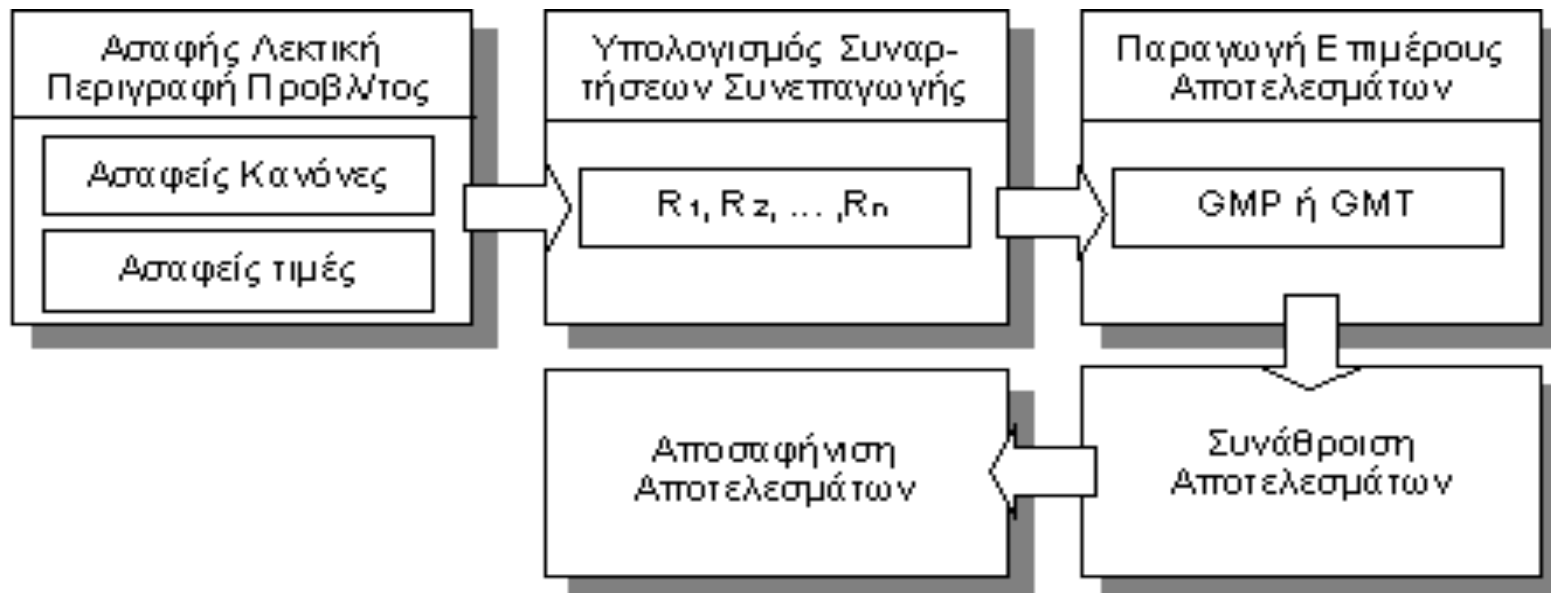
❖ και έστω συλλογιστική διαδικασία GMP (δηλαδή γνωστό το A' ως τιμή του x και ζητούμενο το B' ως τιμή του y), τα ασαφή σύνολα A και B συνδυάζονται με κάποιον από τους τελεστές συνεπαγωγής και παράγουν τη σχέση συνεπαγωγής $R(x,y)$.

❖ Από την $R(x,y)$ μέσω σύνθεσης (έστω *max-min* σύνθεση) με το A' προκύπτει η άγνωστη ποσότητα B' :

$$B' = A' \circ R(x,y)$$

Ασαφής Συλλογιστική

- ❖ Εξαγωγή συμπερασμάτων με χρήση ασαφών κανόνων.
- ❖ Τέσσερα στάδια:
 - ❑ Υπολογισμός της συνάρτησης συνεπαγωγής για κάθε εμπλεκόμενο κανόνα.
 - ❑ Παραγωγή επιμέρους αποτελεσμάτων μέσω κάποιας συλλογιστικής διαδικασίας.
 - ❑ Συνάθροιση των επιμέρους αποτελεσμάτων.
 - ❑ Αποσαφήνιση αποτελεσμάτων.



Παράδειγμα Προβλήματος Ασαφούς Συλλογιστικής (1/2)

❖ Έστω σύστημα που ρυθμίζει τη δόση D μιας φαρμακευτικής ουσίας που πρέπει να χορηγηθεί σε ασθενή, με βάση τη θερμοκρασία του T .

❖ Έστω ότι το σύστημα βασίζεται στους εξής δύο ασαφείς κανόνες:

K_1 : if T is HIGH then D is HIGH

K_2 : if T is LOW then D is LOW

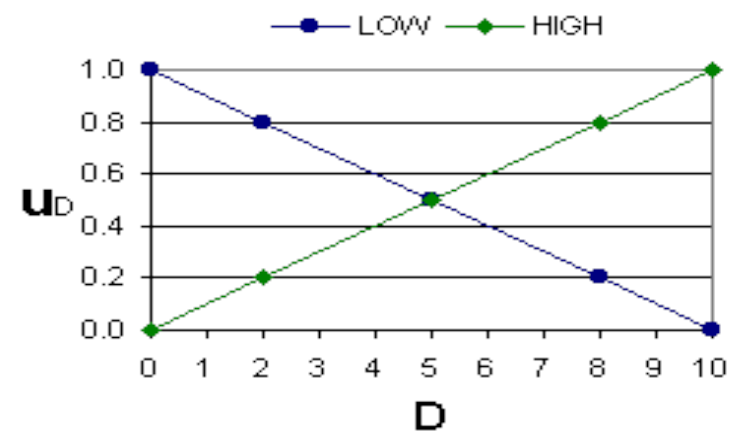
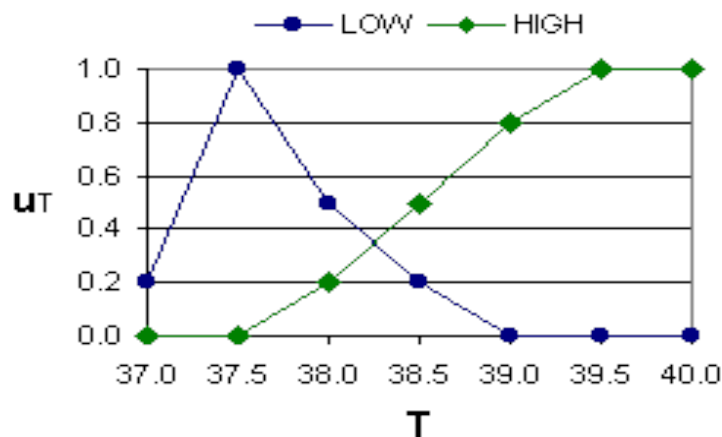
❖ Δίνονται επίσης τα ασαφή σύνολα HIGH και LOW για τα μεγέθη T και D :

$T_{LOW} = \{ 0.2/37, 1/37.5, 0.5/38, 0.2/38.5, 0/39, 0/39.5, 0/40 \}$

$T_{HIGH} = \{ 0/37, 0/37.5, 0.2/38, 0.5/38.5, 0.8/39, 1/39.5, 1/40 \}$

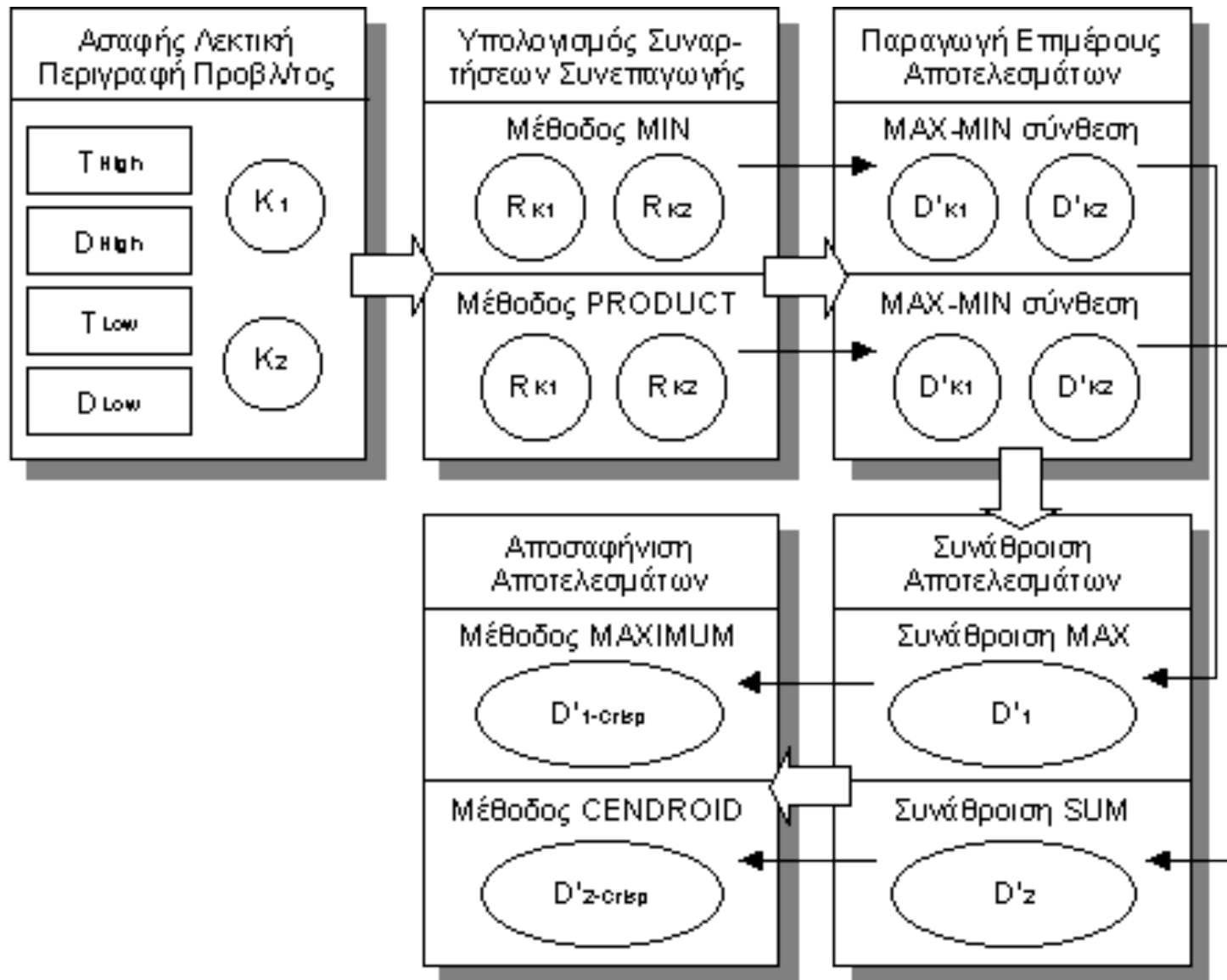
$D_{LOW} = \{ 1/0, 0.8/2, 0.5/5, 0.2/8, 0/10 \}$

$D_{HIGH} = \{ 0/0, 0.2/2, 0.5/5, 0.8/8, 1/10 \}$



❖ Αν $T'=38.5$, να υπολογιστεί η τιμή του D' με συλλογιστική διαδικασία GMP.

Παράδειγμα Προβλήματος Ασαφούς Συλλογιστικής (2/2)



Βήμα A1: Υπολογισμός συνάρτησης συνεπαγωγής

Μέθοδος MIN

❖ 2 κανόνες: δύο τελεστές συνεπαγωγής, οι R_{K1} και R_{K2} .

□ Χρησιμοποιείται ο τελεστής συνεπαγωγής Mandani min (ή απλά MIN).

K_1 : if T is HIGH then D is HIGH

K_2 : if T is LOW then D is LOW

❖ Έστω ο K_1 . Κατασκευάζεται ο πίνακας αριστερά:

R_{K1}	D	0	2	5	8	10
T		0	0.2	0.5	0.8	1
37.0	0	0	0	0	0	0
37.5	0	0	0	0	0	0
38.0	0.2	0	0.2	0.2	0.2	0.2
38.5	0.5	0	0.2	0.5	0.5	0.5
39.0	0.8	0	0.2	0.5	0.8	0.8
39.5	1	0	0.2	0.5	0.8	1
40.0	1	0	0.2	0.5	0.8	1

R_{K2}	D	0	2	5	8	10
T		1	0.8	0.5	0.2	0
37.0	0.2	0.2	0.2	0.2	0.2	0
37.5	1	1	0.8	0.5	0.2	0
38.0	0.5	0.5	0.5	0.5	0.2	0
38.5	0.2	0.2	0.2	0.2	0.2	0
39.0	0	0	0	0	0	0
39.5	0	0	0	0	0	0
40.0	0	0	0	0	0	0

□ Κάθε κελί του εσωτερικού πίνακα περιέχει το $\min(u_{T^{HIGH}}, u_{D^{HIGH}})$ για τα T και D της γραμμής και στήλης στην οποία βρίσκεται.

□ Όμοια προκύπτει και η $R_{K2}(T_{LOW}, D_{LOW})$ για τον κανόνα K_2 (πίνακας δεξιά)

❖ **Γενίκευση:** αν N εκφράσεις στο if τμήμα τότε προκύπτει πίνακας N+1 διαστάσεων.

Βήμα A2: Παραγωγή επιμέρους αποτελεσμάτων (1/2)

❖ Με εφαρμογή της συλλογιστικής διαδικασίας GMP:

□ Κανόνας K_1 : $D'_{K1} = T' \circ R_{K1}(T_{HIGH}, D_{HIGH})$

□ Κανόνας K_2 : $D'_{K2} = T' \circ R_{K2}(T_{LOW}, D_{LOW})$

❖ Απαιτείται η γραφή της θερμοκρασίας $T' = 38.5$ σε μορφή ασαφούς συνόλου, δηλαδή:

$$T' = 38.5 = \{ 0/37, 0/37.5, 0/38, 1/38.5, 0/39, 0/39.5, 0/40 \}$$

❖ Χρησιμοποιείται η μέθοδος σύνθεσης (\circ) max-min (η συνηθέστερη περίπτωση).

❖ Τεχνική όμοια με πολλαπλασιασμό πινάκων: χρησιμοποιείται min αντί πολλαπλασιασμού και max αντί πρόσθεσης.

❖ 1^{ος} πίνακας το ασαφές σύνολο T' (1x7) και 2^{ος} ο αριστερά του βήματος A1 (7x5)

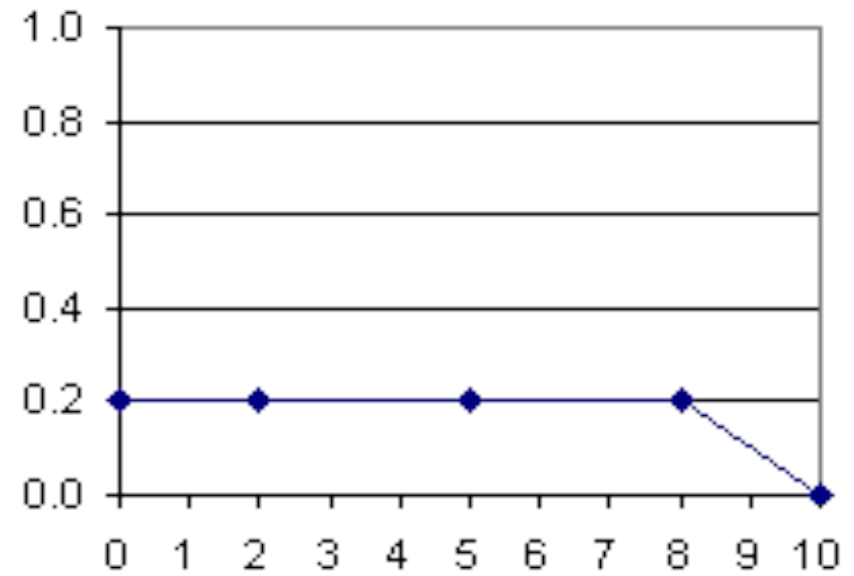
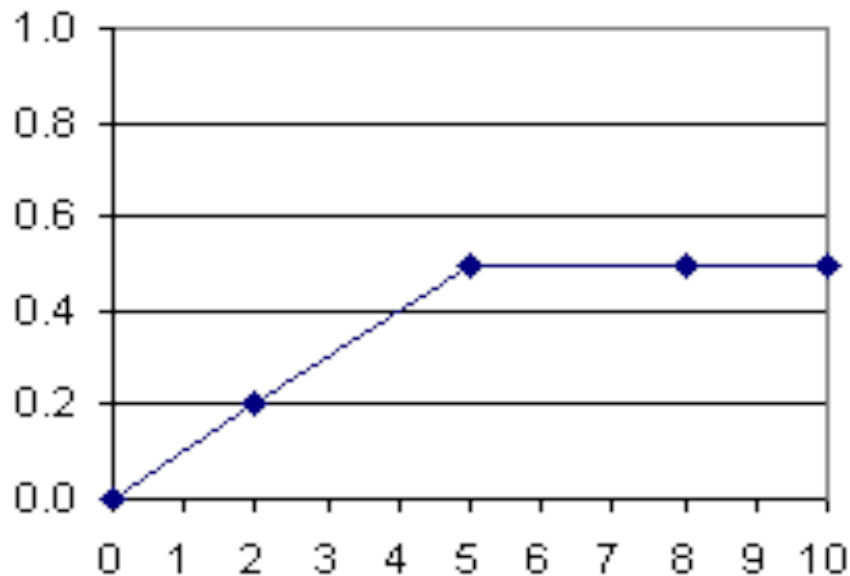
❖ Το αποτέλεσμα θα είναι ένας πίνακας 1x5 που θα αποτελεί και την ποσότητα D'_{K1} .

$$D'_{K1} = [0/37, 0/37.5, 0/38, 1/38.5, 0/39, 0/39.5, 0/40] \circ \begin{array}{|c|c|c|c|c|c|} \hline T & D & 0 & 2 & 5 & 8 & 10 \\ \hline 37 & & 0 & 0 & 0 & 0 & 0 \\ \hline 37.5 & & 0 & 0 & 0 & 0 & 0 \\ \hline 38 & & 0 & 0.2 & 0.2 & 0.2 & 0.2 \\ \hline 38.5 & & 0 & 0.2 & 0.5 & 0.5 & 0.5 \\ \hline 39 & & 0 & 0.2 & 0.5 & 0.8 & 0.8 \\ \hline 39.5 & & 0 & 0.2 & 0.5 & 0.8 & 1 \\ \hline 40 & & 0 & 0.2 & 0.5 & 0.8 & 1 \\ \hline \end{array} \Rightarrow$$

$$D'_{K1} = \{ 0/0, 0.2/2, 0.5/5, 0.5/8, 0.5/10 \}$$

Βήμα A2: Παραγωγή επιμέρους αποτελεσμάτων (2/2)

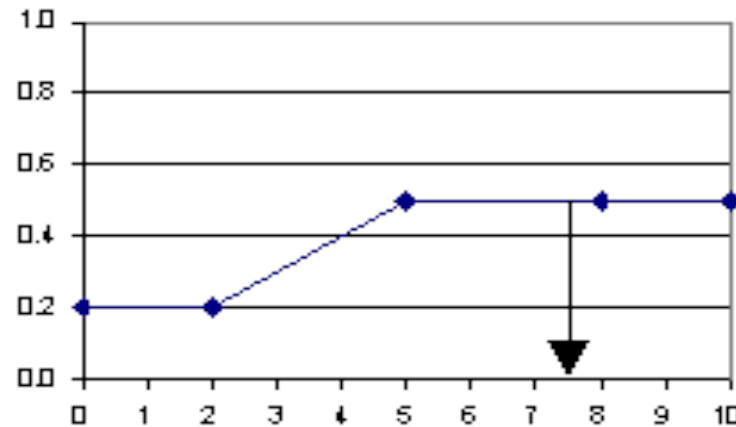
- ❖ Όμοια, ο κανόνας K_2 δίνει: $D'_{K_2} = \{ 0.2/0, 0.2/2, 0.2/5, 0.2/8, 0/10 \}$
- ❖ D'_{K_1} (αριστερά) και D'_{K_2} (δεξιά).



Βήμα A3: Συνάθροιση αποτελεσμάτων

Μέθοδος MAX

- ❖ Υπολογίζει τη συνδυασμένη έξοδο των κανόνων παίρνοντας τη μέγιστη τιμή συγγένειας από τις παραμέτρους εξόδου κάθε κανόνα, σημείο προς σημείο (*pointwise maximum* - $\max_{p/w}$).



- ❖ Δεδομένου ότι έχει υπολογιστεί:

$$D_1'_{k1} = \{ 0/0, 0.2/2, 0.5/5, 0.5/8, 0.5/10 \}$$

$$D_1'_{k2} = \{ 0.2/0, 0.2/2, 0.2/5, 0.2/8, 0/10 \}$$

η συνάθροισή τους κατά *MAX* δίνει

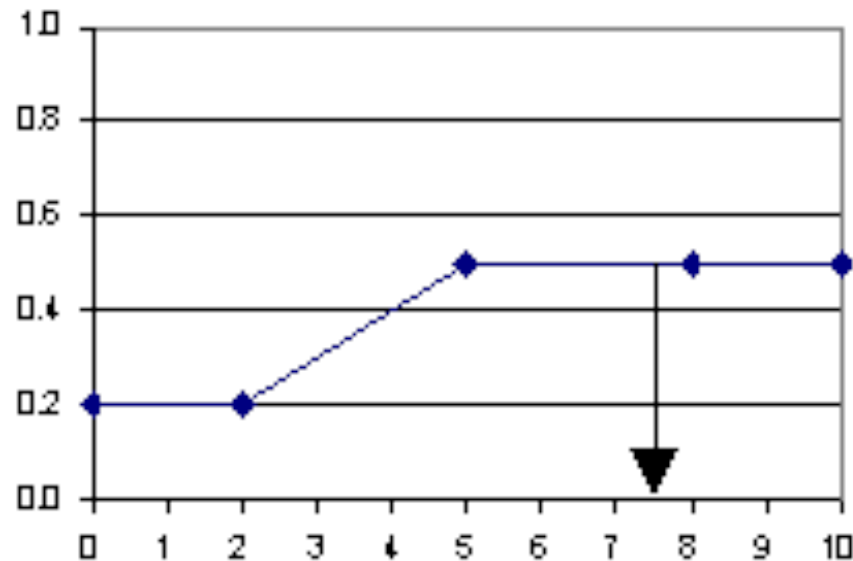
$$D_1' = \{ \max(0,0.2)/0, \max(0.2,0.2)/2, \max(0.5,0.2)/5, \max(0.5,0.2)/8, \max(0.5,0)/10 \} =$$

$$= \{ 0.2/0, 0.2/2, 0.5/5, 0.5/8, 0.5/10 \}$$

Βήμα A4: Αποσαφήνιση

❖ Μέθοδος αποσαφήνισης MAXIMUM

- ❑ διακριτή τιμή: μέγιστη τιμή συγγένειας του τελικού αποτελέσματος.
- ❑ με average-of-maxima αποσαφήνιση: $D_1 = (5+8+10)/3 = 7.7$



Βήμα Β1: Υπολογισμός συνάρτησης συνεπαγωγής

Μέθοδος PRODUCT

❖ δύο κανόνες: δύο τελεστές συνεπαγωγής, οι R_{K1} και R_{K2} .

Έστω ο τελεστής συνεπαγωγής Larsen Product (ή απλά PRODUCT).

K_1 : if T is HIGH then D is HIGH

K_2 : if T is LOW then D is LOW

❖ **Κανόνας K_1** . Κατασκευάζεται ο πίνακας αριστερά:

R_{K1}	D	0	2	5	8	10
T		0	0.2	0.5	0.8	1
37	0	0	0	0	0	0
37.5	0	0	0	0	0	0
38	0.2	0	0.04	0.1	0.16	0.2
38.5	0.5	0	0.1	0.25	0.4	0.5
39	0.8	0	0.16	0.4	0.64	0.8
39.5	1	0	0.2	0.5	0.8	1
40	1	0	0.2	0.5	0.8	1

R_{K2}	D	0	2	5	8	10
T		1	0.8	0.5	0.2	0
37	0.2	0.2	0.16	0.1	0.04	0
37.5	1	1	0.8	0.5	0.2	0
38	0.5	0.5	0.4	0.25	0.1	0
38.5	0.2	0.2	0.16	0.1	0.04	0
39	0	0	0	0	0	0
39.5	0	0	0	0	0	0
40	0	0	0	0	0	0

□ Κάθε κελί του εσωτερικού πίνακα (σχέση συνεπαγωγής R_{K1}) περιέχει το $(u_{T^{HIGH}} \cdot u_{D^{HIGH}})$ για τα T και D της γραμμής και στήλης στην οποία βρίσκεται.

□ Όμοια προκύπτει και η $R_{K2}(T_{LOW}, D_{LOW})$ για τον κανόνα K_2 (πίνακας δεξιά)

Βήμα B2: Παραγωγή επιμέρους αποτελεσμάτων (1/2)

- ❖ Με εφαρμογή της συλλογιστικής διαδικασίας GMP:
 - ❑ Κανόνας K_1 : $D'_{K1} = T' \circ R_{K1}(T_{HIGH}, D_{HIGH})$
 - ❑ Κανόνας K_2 : $D'_{K2} = T' \circ R_{K2}(T_{LOW}, D_{LOW})$
- ❖ Απαιτείται η γραφή της θερμοκρασίας $T' = 38.5$ σε μορφή ασαφούς συνόλου:

$$T' = 38.5 = \{ 0/37, 0/37.5, 0/38, 1/38.5, 0/39, 0/39.5, 0/40 \}$$
- ❖ Χρησιμοποιείται η μέθοδος σύνθεσης (\circ) max-min (η συνηθέστερη περίπτωση).
- ❖ 1^{ος} πίνακας το ασαφές σύνολο T' (1x7) και 2^{ος} ο εσωτερικός του βήματος B1 (7x5)
- ❖ Το αποτέλεσμα θα είναι ένας πίνακας 1x5 που θα αποτελεί και την ποσότητα D'_{K1} .

$$D'_{K1} = [0/37, 0/37.5, 0/38, 1/38.5, 0/39, 0/39.5, 0/40] \quad \circ$$

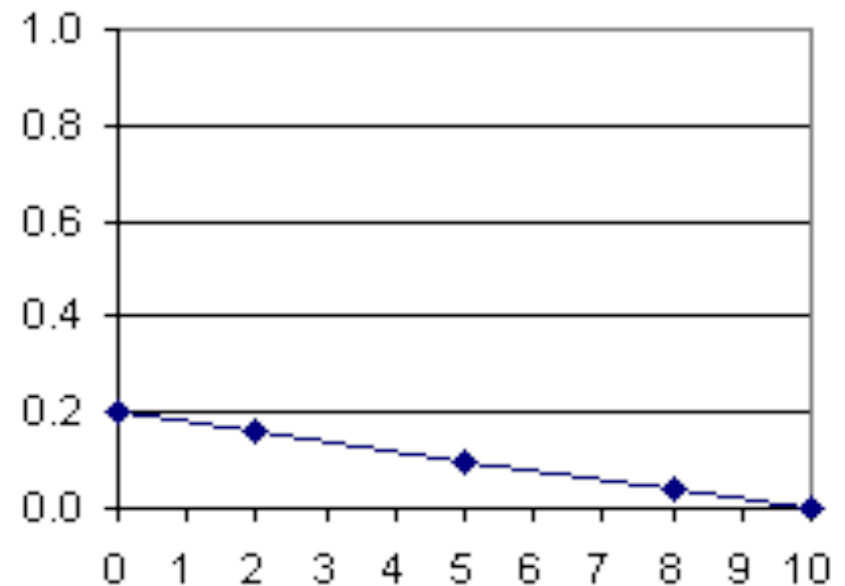
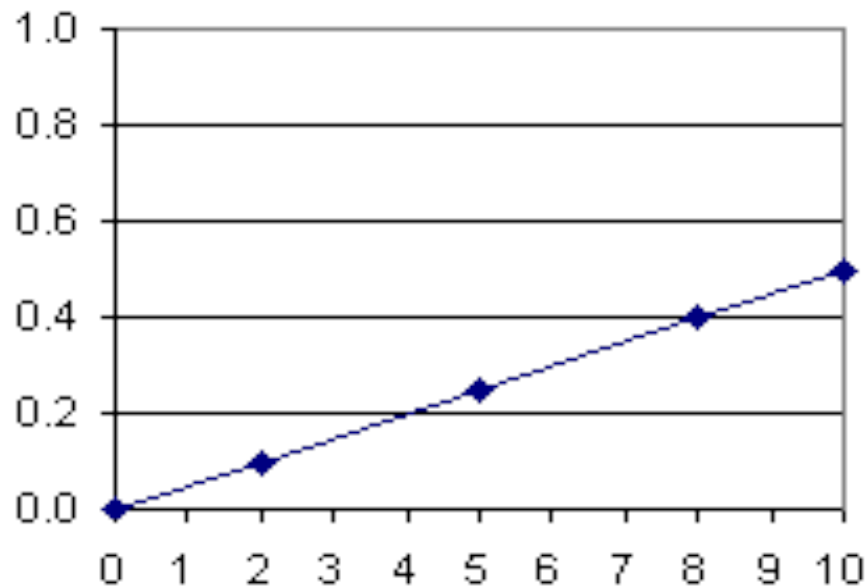
T	D	0	2	5	8	10
37		0	0	0	0	0
37.5		0	0	0	0	0
38		0	0.04	0.1	0.16	0.2
38.5		0	0.1	0.25	0.4	0.5
39		0	0.16	0.4	0.64	0.8
39.5		0	0.2	0.5	0.8	1
40		0	0.2	0.5	0.8	1

\Rightarrow

$$D'_{K1} = \{ 0/0, 0.1/2, 0.25/5, 0.4/8, 0.5/10 \}$$

Βήμα Β2: Παραγωγή επιμέρους αποτελεσμάτων (2/2)

- ❖ Όμοια προκύπτει ότι ο κανόνας K_2 δίνει: $D'_{K_2} = \{ 0.2/0, 0.16/2, 0.1/5, 0.04/8, 0/10 \}$
- ❖ Γραφική απεικόνιση των D'_{K_1} (αριστερά) και D'_{K_2} (δεξιά).



Βήμα Β3: Συνάθροιση αποτελεσμάτων

Μέθοδος SUM

❖ Υπολογίζει τη συνδυασμένη έξοδο των κανόνων παίρνοντας το άθροισμα των τιμών συγγένειας των παραμέτρων εξόδου κάθε κανόνα, σημείο προς σημείο (*pointwise sum* - $\text{sum}_{p/w}$).

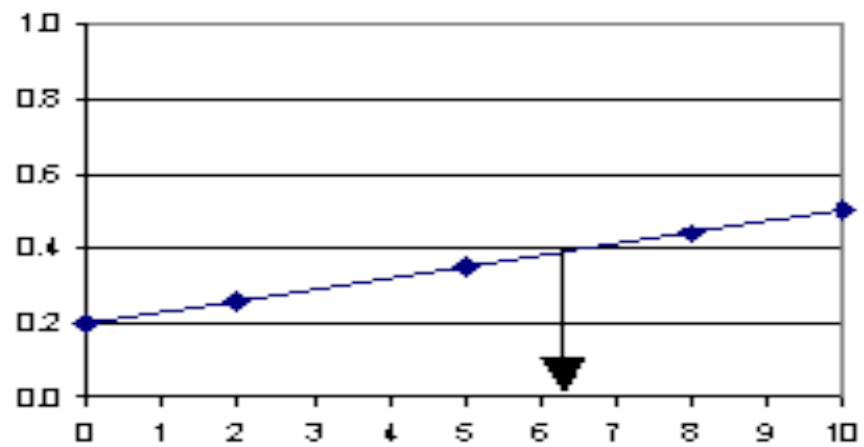
❖ Δεδομένου ότι έχει υπολογιστεί:

□ $D_{2'K1} = \{ 0/0, 0.1/2, 0.25/5, 0.4/8, 0.5/10 \}$

□ $D_{2'K2} = \{ 0.2/0, 0.16/2, 0.1/5, 0.04/8, 0/10 \}$

❖ ...η συνάθροισή τους κατά *MAX* δίνει....

□ $D_2' = \{ (0+0.2)/0, (0.1+0.16)/2, (0.25+0.1)/5, (0.4+0.04)/8, (0.5+0)/10 \} =$
 $\{ 0.2/0, 0.26/2, 0.35/5, 0.44/8, 0.5/10 \}$



Βήμα B4: Αποσαφήνιση

❖ Μέθοδος αποσαφήνισης CENDROID

□ Η διακριτή τιμή είναι αυτή που προκύπτει από το κέντρο βάρους της τελικής συνάρτησης συγγένειας για την ασαφή παράμετρο εξόδου.

□ Το κέντρο βάρους επιφάνειας που ορίζεται από μία συνάρτηση $f(t)$: σχέση (1)

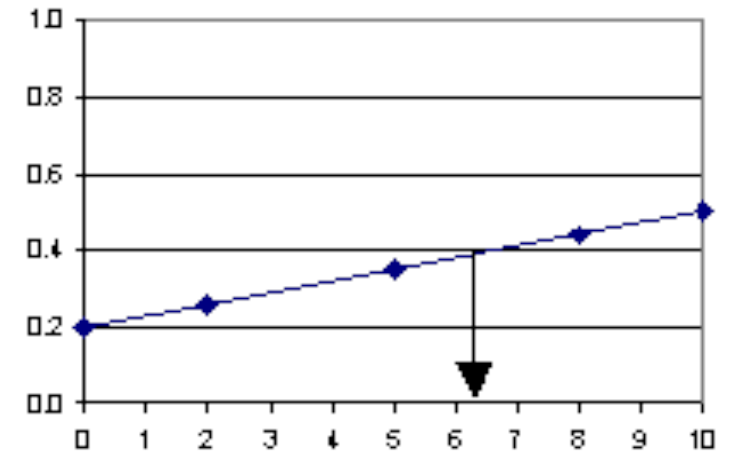
$$(1) \quad t_{\kappa\beta} = \frac{\int_{t1}^{t2} t \cdot f(t) dt}{\int_{t2}^{t1} f(t) dt}$$

$$(2) \quad t_{\kappa\beta} = \frac{\sum_{i=1}^N t_i \cdot u_{OUT}(t_i)}{\sum_{i=1}^N u_{OUT}(t_i)}$$

□ Για διακριτού συνόλου αναφοράς: διακριτό άθροισμα με δειγματοληψία N σημείων (σχ.2).

□ Με CENDROID αποσαφήνιση στα αποτελέσματα της συνάθροισης SUM, προκύπτει:

$$t_{D2'} = \frac{\sum_{i=1}^5 t_i \cdot u_{D2'}(t_i)}{\sum_{i=1}^5 u_{D2'}(t_i)} = \frac{0 \cdot 0.2 + 2 \cdot 0.26 + 5 \cdot 0.35 + 8 \cdot 0.44 + 10 \cdot 0.5}{0.2 + 0.26 + 0.35 + 0.44 + 0.5} = 6.2$$



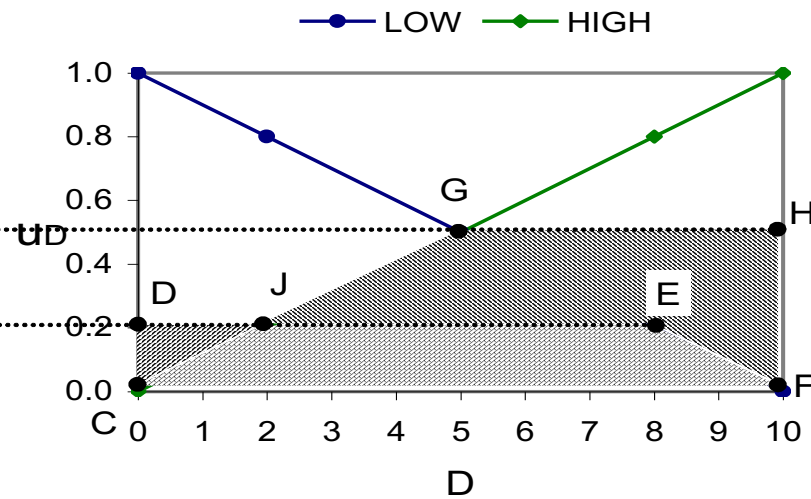
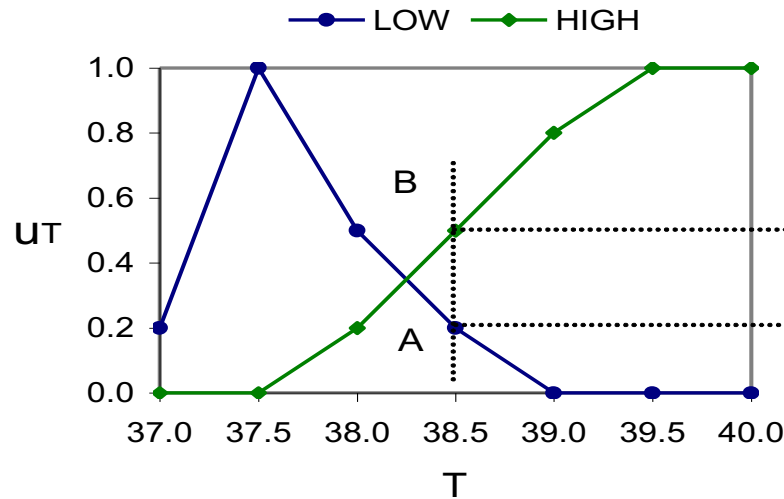
Διαγραμματική Επίλυση (1/2)

- Προϋπόθεση: οι συναρτ. συγγένειας να είναι συνεχείς καμπύλες - όχι ζεύγη $(x, u(x))$
- Πρόβλημα: υπολογισμός της δόσης D' μιας φαρμακευτικής ουσίας με βάση την θερμοκρασία T' και τους ασαφείς κανόνες:

K_1 : if T is HIGH then D is HIGH

K_2 : if T is LOW then D is LOW

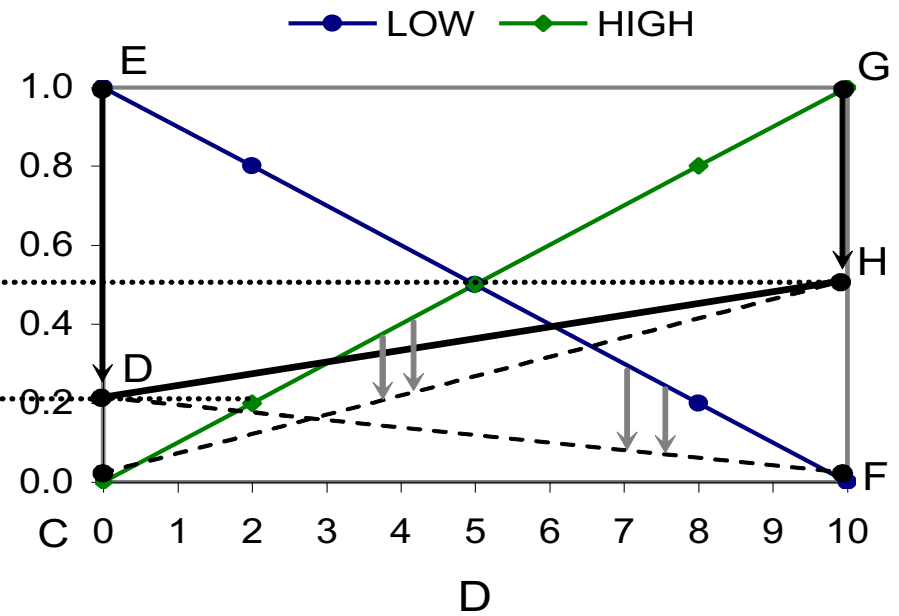
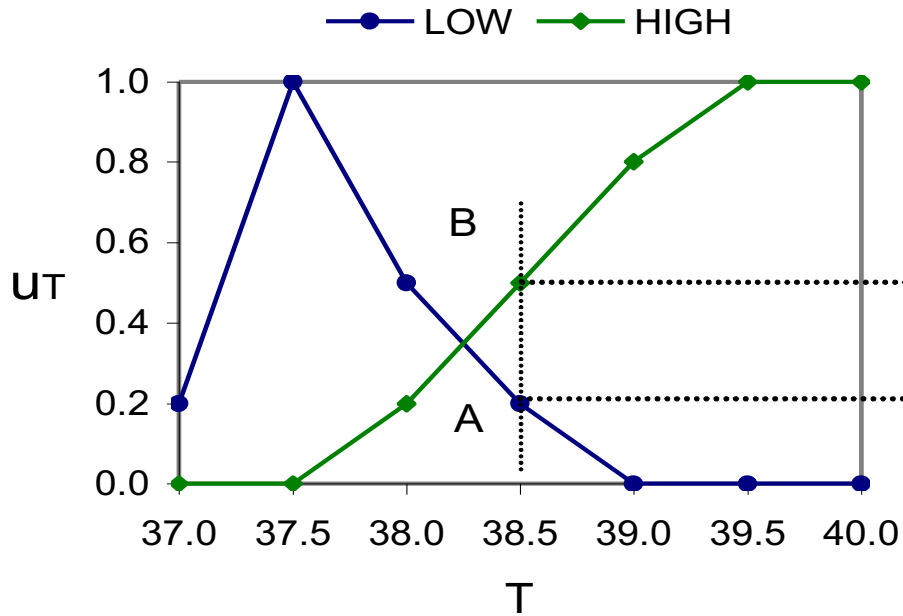
- με μέθοδο MIN για τον υπολογισμό των συναρτήσεων συνεπαγωγής:



- κανόνας K_2 : DEF
- κανόνας K_1 : CGH
- Λύση: αποσαφήνιση στην "καμπύλη" DJGH

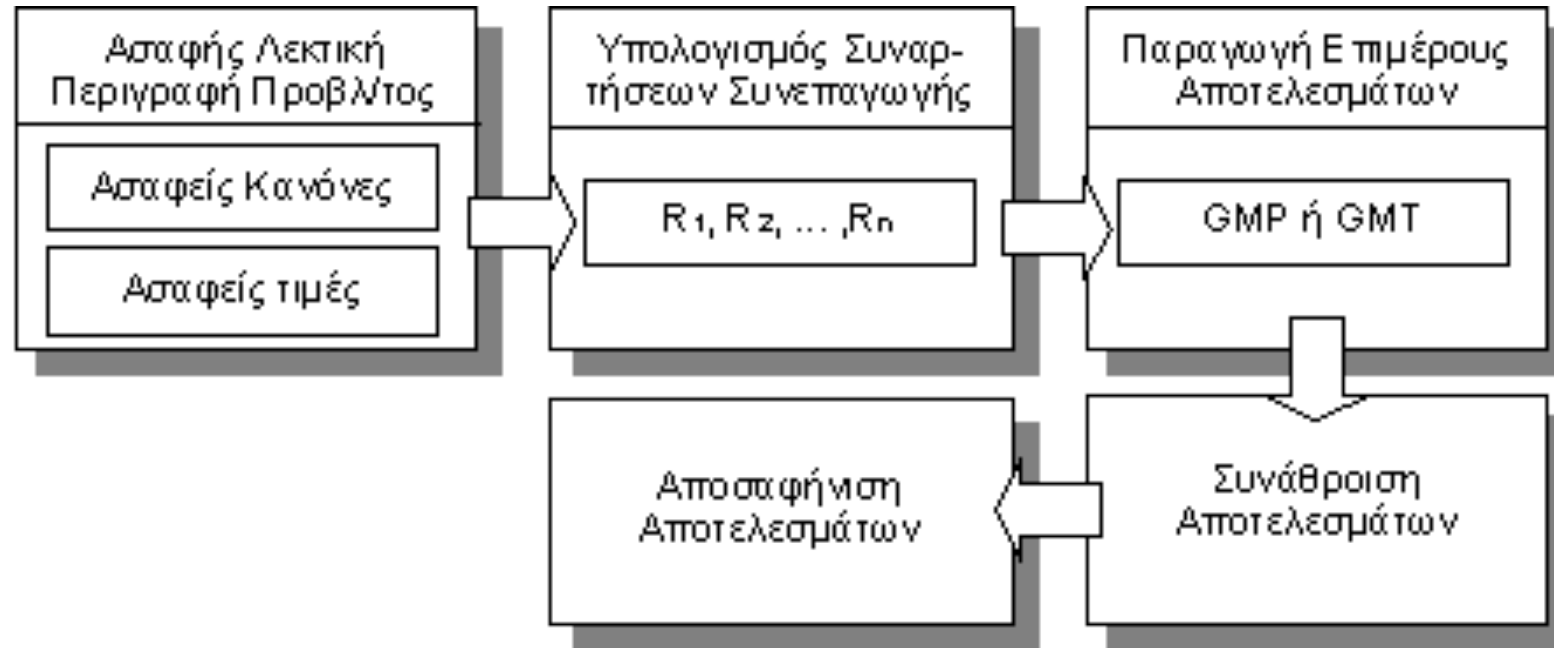
Διαγραμματική Επίλυση (2/2)

- Θεωρώντας μέθοδο PRODUCT για τον υπολογισμό των συναρτήσεων συνεπαγωγής:



- κανόνας K_2 : DF
- κανόνας K_1 : CH
- Λύση: αποσαφήνιση στην "καμπύλη" DH

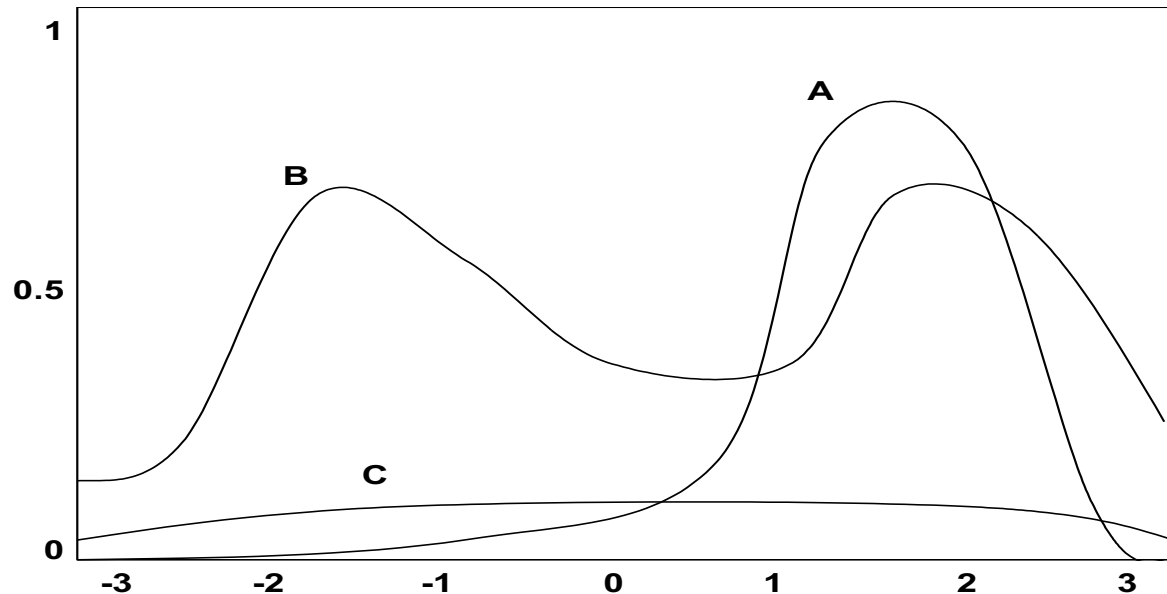
Συστήματα Ασαφούς Συλλογιστικής



- ❖ **Δυσκολότερο σημείο:** επιλογή ασαφών μεταβλητών, των τιμών τους και των κανόνων με τους οποίους θα συνδυαστούν.
- ❖ Συναρτήσεις συγγένειας: χρήση νευρωνικών δικτύων.
- ❖ σημεία προσοχής: επιλογή τελεστή συνεπαγωγής, μεθόδου αποσαφήνισης.

Σταθερότητα/Ποιότητα Ασαφούς Συστήματος

- ❖ **Σταθερότητα:** η ικανότητα να εμφανίζει καλή συμπεριφορά σε όλο το φάσμα τιμών εισόδου.
- ❖ Η μορφή του τελικού αποτελέσματος πολλές φορές δίνει μία καλή ένδειξη για την ποιότητα του συνολικού συστήματος.



Χαρακτηριστικές περιπτώσεις ασαφούς εξόδου.

Παράδειγμα

- a) ύπαρξη ενός "ισχυρού" κανόνα (επιθυμητό χαρακτηριστικό).
- b) δύο κορυφές: αντιφατική συμπεριφορά (απαιτείται βελτίωση του συστήματος κανόνων).
- c) μεγάλο πλατό: το σύστημα των κανόνων είναι ελλιπές.

Εφαρμογές Ασαφούς Λογικής

- ❖ Σύστημα Linkman (ιστορικά η πρώτη εφαρμογή): βιομηχανίες τσιμέντου.
- ❖ Υπόγειος σιδηρόδρομος Sendai στην Ιαπωνία.
- ❖ Φωτογραφικές μηχανές.
- ❖ Πλυντήρια ρούχων.
- ❖ Συσκευές video-camera.
- ❖ Συστήματα πέδησης (fuzzy ABS).
- ❖ Συστήματα ελέγχου λαβής σε ρομποτικούς βραχίονες.
- ❖ Συσκευές κλιματισμού.
- ❖ Βαλβίδες για έλεγχο ροής.
- ❖ Κατανομή καυσίμου ανάλογα με το φάκελο πτήσης σε δεξαμενές πολεμικών αεροσκαφών.
- ❖ Έμπειρα συστήματα με ασαφείς κανόνες.

Συστήματα Γνώσης

- ❖ **Επιδεικνύουν νοήμονα συμπεριφορά** σε συγκεκριμένους τομείς και διαδικασίες, ανάλογη ενός ανθρώπου με ειδικότητα στον τομέα
 - ❑ π.χ. επιστήμονα, τεχνικού, εμπειρογνώμονα
- ❖ **Κωδικοποιούν και χειρίζονται τη γνώση και συλλογιστική** ενός ανθρώπου σε έναν εξειδικευμένο τομέα, με σκοπό την επίλυση προβλημάτων ή την παροχή συμβουλών.
- ❖ **Απαιτούν γνώση**
 - ❑ Αποκτάται μέσω εμπειρίας ή μελέτης.
 - ❑ Πληροφορίες, εμπειρίες, ικανότητες, δεξιότητες
- ❖ **Εμπειρα συστήματα (expert systems)**
 - ❑ Στηριζόταν κυρίως στην εμπειρική γνώση
- ❖ **Χρησιμοποιούνται:**
 - ❑ Από μη-ειδικούς για επίλυση προβλημάτων
 - ❑ Συμβουλευτικά από ειδικούς
- ❖ **Κατηγορίες εφαρμογών:**
 - ❑ Ερμηνεία δεδομένων (π.χ. ηχητικών ή ηλεκτρομαγνητικών σημάτων)
 - ❑ Διάγνωση δυσλειτουργιών (π.χ. βλαβών σε μηχανήματα ή ασθενειών σε ανθρώπους)
 - ❑ Διαμόρφωση σύνθετων αντικειμένων (π.χ. πολύπλοκων υπολογιστικών συστημάτων)

Ανάπτυξη Συστημάτων Γνώσης

❖ Συνεργασία:

☐ Ειδικός του τομέα (domain expert)

- Εξειδικευμένος σε έναν τομέα δραστηριότητας
- Βοηθάει στη μεταφορά γνώσης στο σύστημα

α) δική του εμπειρία

β) κοινή επιστημονική ή τεχνολογική γνώση

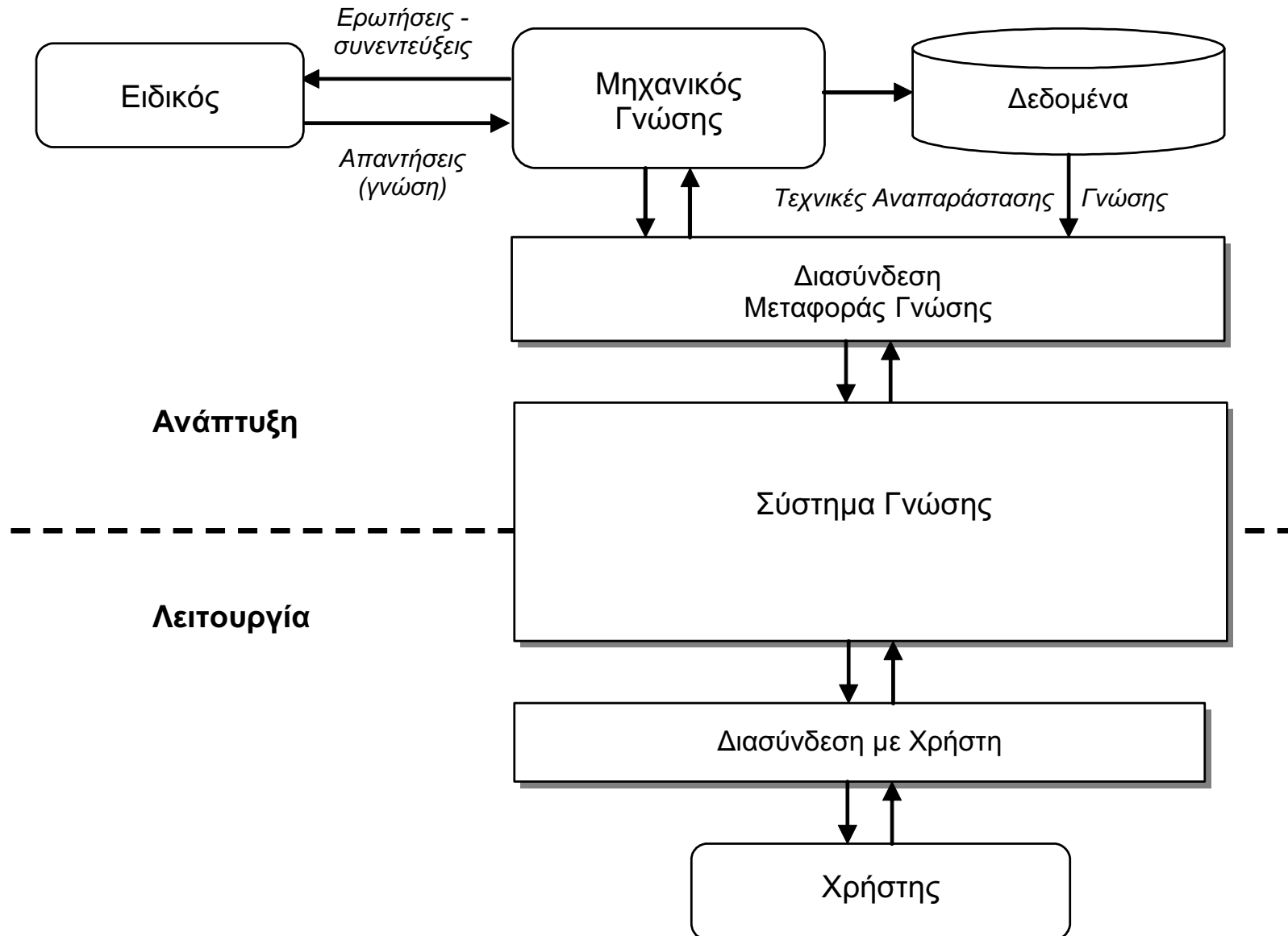
γ) γνώση καταγεγραμμένη σε βάσεις δεδομένων ή έγγραφα

☐ Μηχανικός γνώσης (knowledge engineer)

- Πληροφορικός, ειδικευμένος σε ΤΝ και συστήματα γνώσης
- Συνεργάζεται με τον ειδικό για τη μεταφορά γνώσης στο σύστημα
- Σχεδιάζει τη δομή γνώσης, σχεδιάζει και αναπτύσσει το σύστημα

❖ Τεχνολογία Της Γνώσης (Knowledge Engineering): τομέας της ΤΝ που ασχολείται με την ανάπτυξη συστημάτων γνώσης

Ανάπτυξη και Λειτουργία Συστήματος Γνώσης



Χαρακτηριστικά Συστημάτων Γνώσης

ΣΥΣΤΗΜΑΤΑ ΓΝΩΣΗΣ	ΣΥΜΒΑΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ
Παράσταση και χειρισμός γνώσης σε επίπεδο συμβόλων	Παράσταση και χειρισμός δεδομένων σε επίπεδο αριθμητικών υπολογισμών
Χρήση γλωσσών που πλησιάζουν την ανθρώπινη	Χρήση γλωσσών που βρίσκονται πλησιέστερα στον τρόπο λειτουργίας του Η/Υ
Βάση γνώσης (δεδομένα και εξαγωγή συμπερασμάτων)	Βάση δεδομένων - η γνώση ενσωματώνεται στο πρόγραμμα
Ευχέρεια στην επέκταση και αναθεώρηση της γνώσης	Η αναθεώρηση της υπάρχουσας γνώσης επιβάλλει ευρείας κλίμακας μεταβολές στο πρόγραμμα
Δυνατότητα χειρισμού ασαφούς, αβέβαιης και μη-πλήρους γνώσης	Δυσχέρεια στο χειρισμό ασαφούς, αβέβαιης και μη-πλήρους γνώσης
Δυνατότητα μη μονότονης συλλογιστικής	Δυσχέρεια στη χρήση μη μονότονης συλλογιστικής
Επεξήγηση του δρόμου συλλογισμού	Ανυπαρξία επεξήγησης

Χαρακτηριστικά Εμπείρων Συστημάτων

❖ Επιπλέον χαρακτηριστικά:

ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ	ΣΥΜΒΑΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ
Προσομοιώνουν τον τρόπο επίλυσης ενός προβλήματος	Προσομοιώνουν το ίδιο το πρόβλημα
Χρήση ευριστικών μεθόδων για περιορισμό του χώρου αναζήτησης	Χρήση αλγορίθμων

Πλεονεκτήματα / Μειονεκτήματα Συστήματος Γνώσης

Ισχύει και για τα συστήματα συλλογιστικής των περιπτώσεων

ΑΝΘΡΩΠΟΣ ΕΙΔΙΚΟΣ		ΣΥΣΤΗΜΑ ΓΝΩΣΗΣ	
Μειονεκτήματα	Γνώση διαθέσιμη όταν ο ίδιος είναι παρών	Πλεονεκτήματα	Γνώση πάντα διαθέσιμη.
	Δυσκολία μεταφοράς-αποτύπωσης γνώσης		Ευκολία μεταφοράς-αποτύπωσης γνώσης
	Συναισθηματικές παρορμήσεις		Εργάζεται με συνέπεια
	Η απόδοσή του επηρεάζεται από εξωγενείς παράγοντες		Εργάζεται οπουδήποτε
	Υψηλό κόστος		Χαμηλό κόστος λειτουργίας / Υψηλό κόστος ανάπτυξης
	Υποκειμενικότητα		Αντικειμενικότητα αν η γνώση προέρχεται από πολλούς ειδικούς
Πλεονεκτήματα	Δημιουργικότητα, Ευρύνοια	Μειονεκτήματα	Απουσία έμπνευσης, Περιορισμένο πεδίο σκέψης
	Κοινή λογική		Δυσχέρεια στη μεταφύτευση της κοινής λογικής
	Γνώση των ορίων και δυνατοτήτων τους (μετα-γνώση)		Έλλειψη μετα-γνώσης
	Εκφραστική και λειτουργική επεξήγηση του τρόπου σκέψης τους		Μηχανική επεξήγηση του τρόπου λήψης απόφασης
	Ο έλεγχος της γνώσης γίνεται υποσυνείδητα		Πρέπει η γνώση να ελέγχεται για ορθότητα, πληρότητα και συνέπεια
	Αυτονομία στη μάθηση		Πρέπει να προγραμματιστούν για να μαθαίνουν αυτόματα
Απόκριση σε πραγματικό χρόνο	Δυσκολία απόκρισης σε πραγματικό χρόνο		

Εφαρμογές των Συστημάτων Γνώσης

❖ Διάγνωση (diagnosis).

- ❑ Διάγνωση βλαβών ενός συστήματος βάσει παρατηρήσεων και μετρήσεων.

❖ Πρόγνωση (prognosis-prediction).

- ❑ Πρόβλεψη πιθανών μελλοντικών επιπτώσεων με βάση δεδομένες καταστάσεις.

❖ Εκπαίδευση (instruction).

- ❑ Κατανόηση, αξιολόγηση και διόρθωση απάντησης μαθητών σε εκπαιδευτικά προβλήματα.

❖ Παρακολούθηση καταστάσεων (monitoring).

- ❑ Σύγκριση παρατηρούμενων παραμέτρων με αναμενόμενες καταστάσεις.

❖ Επιδιόρθωση λαθών (repair-remedy).

- ❑ Ανάπτυξη και εκτέλεση σχεδίων (πλάνων) για τη διαχείριση βλαβών.

❖ Ερμηνεία (interpretation).

- ❑ Περιγραφή αντικειμένων και καταστάσεων βάσει δεδομένων από παρατηρήσεις.

❖ Διαμόρφωση (configuration).

- ❑ Ικανοποίηση απαιτήσεων και περιορισμών για τη συναρμολόγηση εξαρτημάτων.

❖ Έλεγχος (control).

- ❑ Έλεγχος της συμπεριφοράς ενός συστήματος. Περιλαμβάνει πολλά από τα παραπάνω.

Γνωστά Συστήματα Γνώσης

❖ DENDRAL

- ❑ Ταυτοποίηση χημικών ενώσεων μέσω φασματικής ανάλυσης.
- ❑ Χρήση ευριστικών κανόνων για περιορισμό του χώρου αναζήτησης.

❖ MYCIN

- ❑ Διάγνωση και θεραπεία της μηνιγγίτιδας και της βακτηριαιμίας.
- ❑ Χρήση συντελεστή βεβαιότητας για τις λύσεις, λόγω αβεβαιότητας απαντήσεων χρήστη.

❖ PROSPECTOR

- ❑ Πρόβλεψη της ακριβούς θέσης ορυκτών κοιτασμάτων αξιοποιώντας γεωλογικά δεδομένα.
- ❑ Χρήση σημασιολογικών δικτύων και δικτύων πιθανοτήτων.

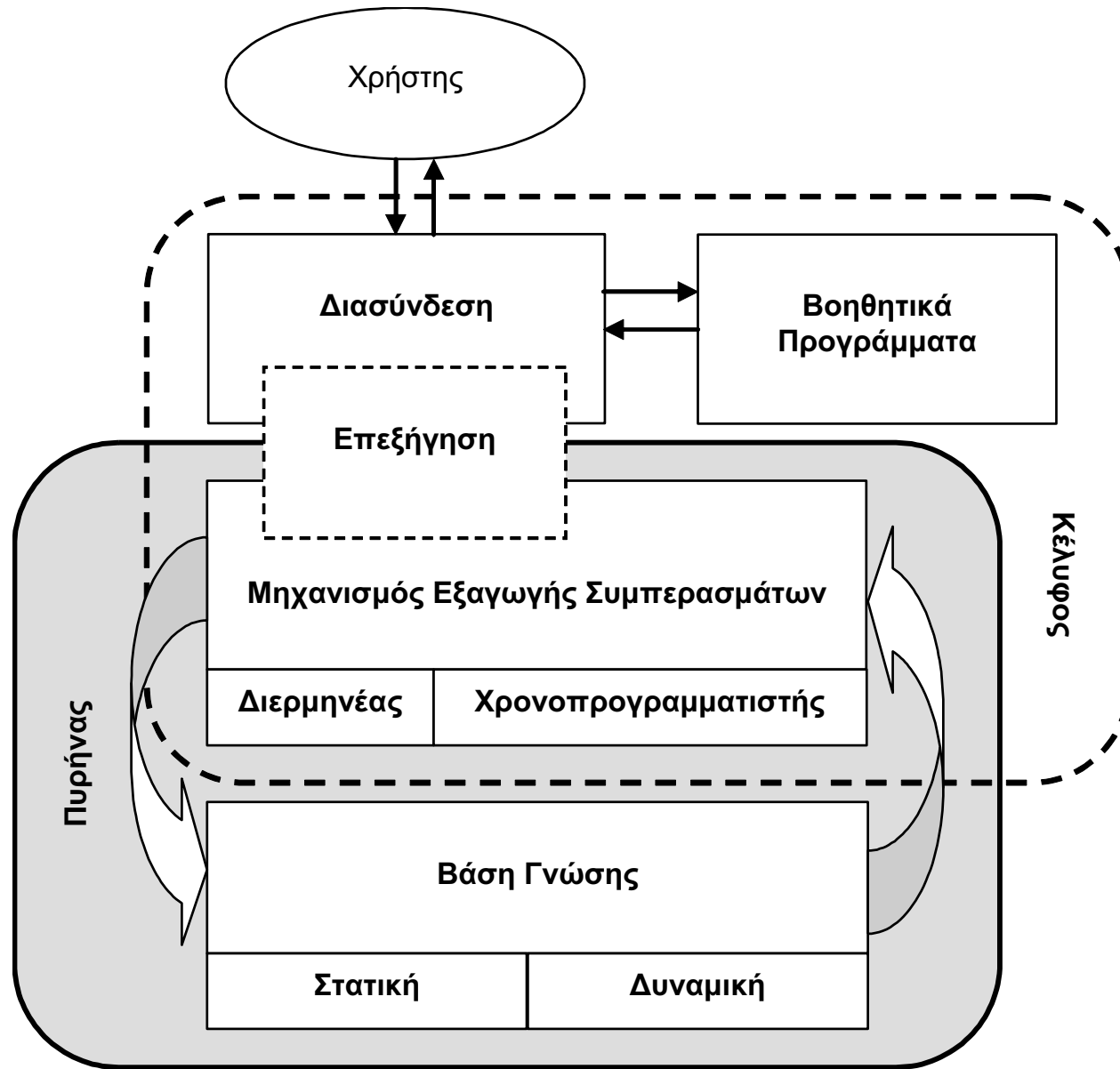
❖ INTERNIST

- ❑ Διάγνωση παθολογικών περιπτώσεων με πολύ μεγάλο αριθμό εναλλακτικών διαγνώσεων.
- ❑ Χρήση ευριστικής συλλογιστικής (απαγωγική) για την πιθανότερη διάγνωση.

❖ XCON

- ❑ Διαμόρφωση υπολογιστών DEC, για να ανταποκρίνονται στις προδιαγραφές του πελάτη.
- ❑ Αναζήτηση κατάλληλου συνδυασμού και χωρικής διάταξη των εξαρτημάτων, με αποφυγή των ασυμβατοτήτων λειτουργίας και διασύνδεσης μεταξύ τους.

Αρχιτεκτονική Συστημάτων Γνώσης (1/2)



Αρχιτεκτονική Συστημάτων Γνώσης (2/2)

❖ Πυρήνας:

- ❑ Βάση γνώσης
- ❑ Μηχανισμό εξαγωγής συμπερασμάτων
- ❑ Ο διαχωρισμός της γνώσης από το μηχανισμό χειρισμού προσφέρει **διαφάνεια**
 - Με αλλαγή της γνώσης, το σύστημα μπορεί να εκτελεί διαφορετικές λειτουργίες

❖ Διασύνδεση

❖ Βοηθητικά προγράμματα

- ❑ π.χ. γραφικά-στατιστικά πακέτα, βάσεις δεδομένων, κτλ.

❖ Κέλυφος (shell)

- ❑ Συνδυασμός της διασύνδεσης με το μηχανισμό εξαγωγής συμπερασμάτων
- ❑ Προέρχεται από την αφαίρεση της βάσης γνώσης από ένα σύστημα γνώσης
- ❑ Αποτελεί εργαλείο ανάπτυξης συστημάτων γνώσης
- ❑ **Κέλυφος εμπείρων συστημάτων** (*expert system shell*)

Βάση Γνώσης

Knowledge Base

- ❖ Περιέχει τη γνώση του συστήματος
- ❖ Υπάρχουν διάφορες μορφές αναπαράστασης γνώσης
 - ❑ π.χ. κανόνες, πλαίσια
- ❖ **Στατική:**
 - ❑ Δε μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.
 - ❑ Αρχικά δεδομένα, διαδικασίες, κανόνες, πλαίσια
 - ❑ Περιγραφή του προβλήματος και των γνωσιολογικών διαδικασιών επίλυσής του
- ❖ **Δυναμική:**
 - ❑ Χώρος εργασίας (**working memory**).
 - ❑ Μερικά συμπεράσματα που δημιουργούνται κατά την εκτέλεση του προγράμματος
 - ❑ Τελική λύση του προβλήματος

Μηχανισμός Εξαγωγής Συμπερασμάτων

Inference Engine

- ❖ Χειρισμός της βάσης γνώσης και εξαγωγή συμπερασμάτων
- ❖ **Διερμηνέας (interpreter):**
 - ❑ Χειρισμός υπάρχουσας γνώσης και παραγωγή νέας
 - ❑ Εφαρμογή της συλλογιστικής (reasoning) και εκτέλεση των κανόνων
- ❖ **Χρονοπρογραμματιστής (scheduler):**
 - ❑ Αποφασίζει πότε και με ποια σειρά θα χρησιμοποιηθούν οι κανόνες
 - ❑ Επιλύει το πρόβλημα της **συγκρούσεως (conflict)**
 - ❑ Στρατηγικές επίλυσης συγκρούσεων:
 - Ποιοι κανόνες είναι υποψήφιοι για να λύσουν το πρόβλημα
 - Με ποιον τρόπο θα γίνει η επιλογή
 - Ποιος από τους κανόνες αυτούς τελικά θα επιλεγεί
 - Τι θα γίνει με τους υπόλοιπους κανόνες
 - ❑ Μετα-κανόνες.

Διασύνδεση

- ❖ Δημιουργεί φιλικό περιβάλλον για την επικοινωνία του χρήστη με το σύστημα
- ❖ Τελικός χρήστης (end user)
 - ❑ Χρησιμοποιεί το σύστημα γνώσης
 - ❑ Εύκολη διατύπωση ερωτήσεων με τη βοήθεια γραφικών ευκολιών (π.χ. μενού)
 - ❑ Ανάγνωση απαντήσεων με τη βοήθεια γραφικών απεικονίσεων
- ❖ Ειδικός (expert) ή/και μηχανικός γνώσης (knowledge engineer)
 - ❑ Προσθήκη ή αλλαγή γνώσης
 - ❑ Έλεγχος συμφωνίας της νέας γνώσης με την παλιά (consistency check)

Μηχανισμός Επεξήγησης

❖ Ερωτήσεις του χρήστη σχετικά με:

- ❑ Τους σκοπούς των ερωτήσεων
- ❑ Την πορεία του συλλογισμού

❖ Ο **μηχανισμός επεξήγησης** αλληλεπιδρά με το μηχανισμό εξαγωγής συμπερασμάτων

- ❑ Η πορεία της συλλογιστικής συνδέεται άμεσα με τον τρόπο εκτέλεσης των κανόνων.

❖ Ερωτήσεις:

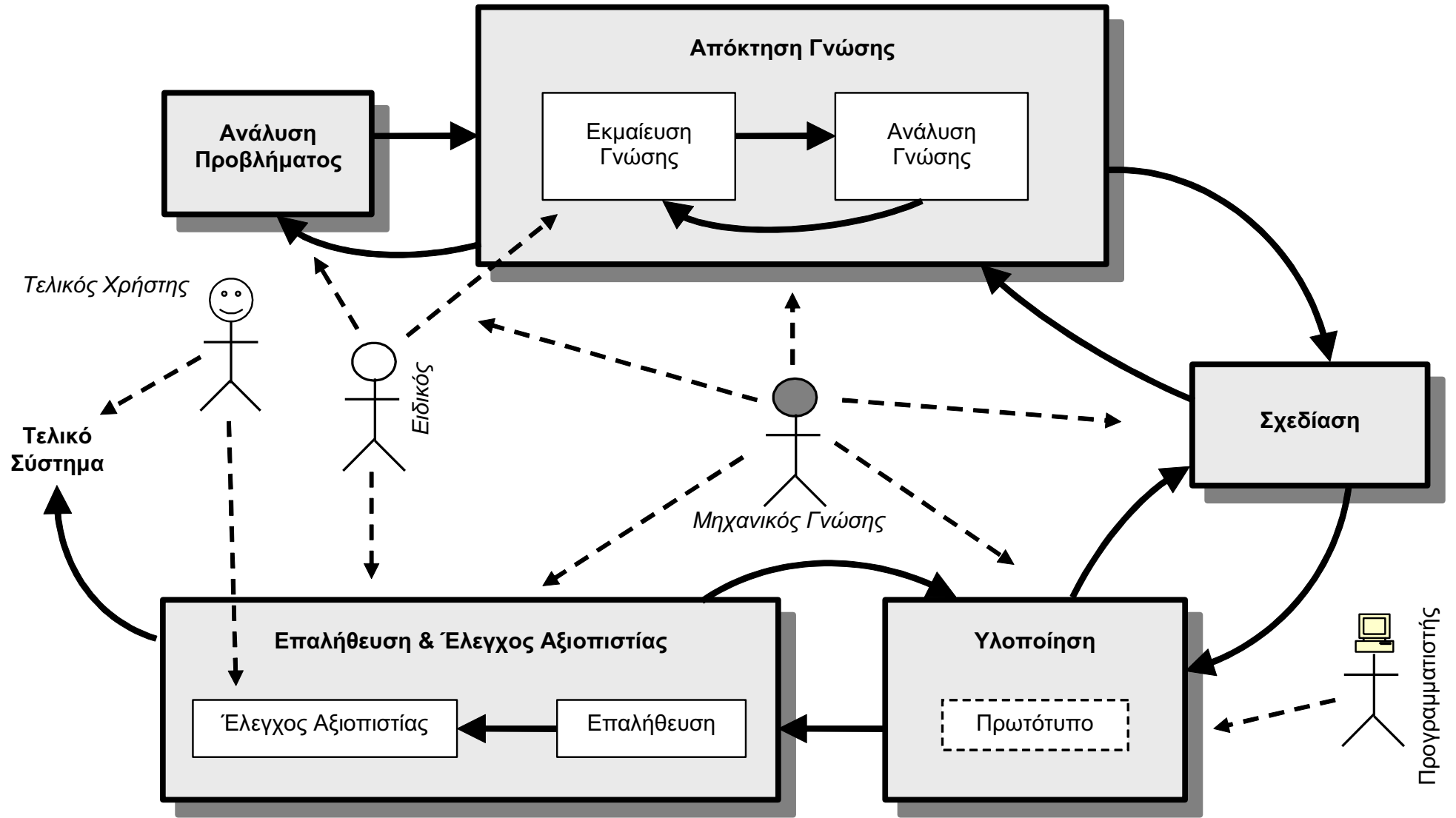
- ❑ **Πώς (how)** κατέληξε σε ένα συμπέρασμα

- Κρατάει πληροφορίες σχετικά με την αποδεικτική διαδικασία
- Παραθέτει τους κανόνες που ενεργοποιήθηκαν σε κάθε κύκλο λειτουργίας και οδήγησαν στην απόδειξη της τρέχουσας απάντησης

- ❑ **Γιατί (why)** ζητά κάποια πληροφορία από το χρήστη

- Ποιοι κανόνες έχουν στην υπόθεσή τους την τρέχουσα πληροφορία
- Επιστρέφει την κατοπινή αλυσίδα συλλογισμών που θα προκαλέσει η ενεργοποίηση αυτών των κανόνων

Διαδικασία Ανάπτυξης Συστημάτων Γνώσης



Ανάλυση Προβλήματος

- ❖ Προσδιορίζεται η επιθυμητή λύση του προβλήματος
- ❖ Κυριότερα ζητήματα:
 - ❑ Είναι το πρόβλημα **κατάλληλο** για επίλυση από σύστημα γνώσης ή συμβατικό πρόγραμμα;
 - ❑ Υπάρχουν έτοιμες μελέτες περιπτώσεων επίλυσης του προβλήματος (**case-studies**);
 - ❑ Ποια είναι τα **οφέλη** από την κατασκευή του συστήματος γνώσης;

Απόκτηση της Γνώσης (*knowledge acquisition*)

- ❖ Ο μηχανικός της γνώσης
 - ❑ Εκμαιεύει από τον ειδικό τη γνώση του πάνω στο πρόβλημα (*knowledge elicitation*)
 - ❑ Μοντελοποιεί (*knowledge analysis & modeling*) τη γνώση σε κάποια ενδιάμεση μορφή αναπαράστασης
- ❖ Εκμαίευση γνώσης (**knowledge elicitation**)
 - ❑ Απαιτεί συνεχή επικοινωνία ανάμεσα στο μηχανικό γνώσης και τον ειδικό.
 - ❑ Εκτός από τις κλασικές μεθόδους, υπάρχουν:
 - **Ημι-αυτόματες μέθοδοι:** π.χ. TEIRESIAS, OPAL, κλπ
 - **Αυτόματες μέθοδοι:** τεχνικές μηχανικής μάθησης
- ❖ Η απόκτηση γνώσης και η παρουσία ειδικού είναι απαραίτητη ακόμα και όταν η γνώση δεν είναι εμπειρική.
 - ❑ *Συλλογιστική των μοντέλων:* Ο μηχανικός γνώσης δεν είναι πάντα δυνατό να μπορεί να ερμηνεύσει τα εγχειρίδια επιστημονικής γνώσης.
 - ❑ *Συλλογιστική των περιπτώσεων:* Ο ειδικός καθορίζει τη σπουδαιότητα των χαρακτηριστικών, τη μέθοδο δεικτοδότησης, τη μέθοδο προσαρμογής των λύσεων, κλπ.

Μοντελοποίηση Γνώσης

Knowledge Analysis & Modeling

- ❖ **Ανάλυση** της γνώσης από το μηχανικό με σκοπό τη δημιουργία ενός **μοντέλου** της γνώσης.
 - ❑ Η αναπαράσταση της γνώσης γίνεται με διάφορες ημιδομημένες μορφές αναπαράστασης.
- ❖ Υπάρχουν μεθοδολογίες που τυποποιούν τη μοντελοποίηση της γνώσης (KADS).
- ❖ Το μοντέλο βοηθά στο να αποκαλυφθούν ατέλειες, ασάφειες και ελλείψεις στη γνώση

Σχεδίαση

- ❖ Προσδιορίζονται:
 - ❑ Η μορφή της αναπαράστασης της γνώσης και η συλλογιστική
 - ❑ Το εργαλείο για την ανάπτυξη του συστήματος γνώσης.
- ❖ Παράγεται η αρχιτεκτονική του συστήματος
 - ❑ Λειτουργικές υπομονάδες (modules) του συστήματος
 - ❑ Λειτουργικότητα της κάθε υπομονάδας
 - ❑ Αλληλεξαρτήσεις υπομονάδων

Υλοποίηση

- ❖ Κωδικοποιείται το μοντέλο της γνώσης χρησιμοποιώντας εργαλεία ανάπτυξης
- ❖ Αρχικά αναπτύσσεται ένα πρωτότυπο σύστημα.
 - ❑ Επιδεικνύεται στον ειδικό και σε μία μικρή ομάδα χρηστών
 - ❑ Καθοδηγεί στη συνέχεια την ανάπτυξη, ή
 - ❑ Οδηγεί σε επανασχεδιασμό όταν δεν ικανοποιεί τις απαιτήσεις που τέθηκαν στην αρχή.
 - ❑ Επαληθεύει τη γνώση που αποκτήθηκε από τον ειδικό και μοντελοποιήθηκε από το μηχανικό γνώσης.
- ❖ Το βάθος της γνώσης πρέπει να είναι μεγάλο.
 - ❑ Πρέπει να μπορεί επιλύσει πλήρως μερικά από τα προβλήματα για τα οποία προορίζεται το σύστημα γνώσης.
- ❖ Το εύρος της γνώσης δε χρειάζεται να είναι μεγάλο.
 - ❑ Δεν είναι αναγκαίο να αντιμετωπίζει πολλές περιπτώσεις.

Επαλήθευση και Έλεγχος Αξιοπιστίας

❖ Επαλήθευση (verification):

- ❑ Έλεγχος της συμβατότητας του συστήματος με τις αρχικές προδιαγραφές.
- ❑ Επιβεβαίωση της συνέπειας και πληρότητας της κωδικοποίησης της γνώσης που περιέχεται στο σύστημα

❖ Έλεγχος αξιοπιστίας (validation):

- ❑ Επιβεβαίωση της ορθότητας και γενικότητας της γνώσης που περιέχει το σύστημα
- ❑ Το σύστημα επιλύει ένα σύνολο από υποδειγματικές περιπτώσεις (test cases).
 - Διαφορετικές από αυτές που χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος
 - Εξασφαλίζεται η **ευρωστία (robustness)** σε μη-προσδοκώμενα δεδομένα
- ❑ Οι λύσεις συγκρίνονται με λύσεις που δόθηκαν από διάφορους ειδικούς του τομέα.

Εκμαίευση Γνώσης

Knowledge Elicitation

- ❖ Διαδικασία απόκτησης (εκμαίευσης) της γνώσης από άτομα που θεωρούνται "ειδικοί" στο συγκεκριμένο τομέα γνώσης (*domain experts*).
- ❑ Π.χ. επιστήμονες, τεχνικοί ή εμπειρογνώμονες.
- ❑ Θεωρείται το πιο δύσκολο-αμφίβολο βήμα στην ανάπτυξη ενός συστήματος γνώσης.
- ❖ **Ειδικός** είναι το άτομο που έχει ειδική γνώση ή ικανότητα πάνω σε ένα θέμα.
- ❖ **Γνώση:** Κατανόηση του κόσμου η οποία αποκτάται μέσω εμπειρίας ή μελέτης.
- ❑ Πληροφορίες, εμπειρίες, ικανότητες, δεξιότητες
- ❑ Είδη γνώσης:
 - Αντικείμενα, γεγονότα, διαδικασίες, κτλ.,
 - Βαθιά-επιφανειακή γνώση (deep-shallow knowledge).
 - Ρητή και άρρητη γνώση (explicit-tacit knowledge)

Προβλήματα στην Εκμαίευση της Γνώσης

❖ Παράδοξο της ειδίκευσης

❑ Όσο πιο πολύ ισχυρίζεται κάποιος ότι είναι ειδικός σε κάποιο θέμα, τόσο πιο δύσκολη είναι η ανταλλαγή πληροφοριών μαζί του.

❖ Ευσεβής πόθος (wishful thinking)

❑ Ο ειδικός εκφράζει το **τι θα έπρεπε να γίνεται** και όχι το **τι πραγματικά γίνεται**.

❖ Κατάλληλο υπόβαθρο γνώσης του μηχανικού γνώσης.

❖ Έλλειψη χρόνου

❖ Ο μηχανικός γνώσης πρέπει να διασφαλίσει ότι ικανοποιούνται οι **στόχοι** της **συνέντευξης**.

❖ Αμεροληψία του μηχανικού γνώσης

❖ Ανεπιτήδειος έμπειρος (inexpert expert)

❖ Απροθυμία του ειδικού να μεταδώσει γνώση

❖ Ο ειδικός μπορεί να μην είναι εξοικειωμένος στη διαδικασία της εκμαίευσης γνώσης

❖ Αδυναμία του ειδικού να **θυμηθεί** ακριβώς τη **ροή** των γεγονότων σε παλιές περιπτώσεις

❖ Επικοινωνιακά προβλήματα του ειδικού ή/και του μηχανικού γνώσης

❖ Υποκειμενικότητα του ειδικού

Χρήση ενός μόνο Ειδικού

Πλεονεκτήματα	Μειονεκτήματα
<ul style="list-style-type: none"> ❖ Προτιμητέο όταν αναπτύσσεται ένα απλό σύστημα ΔΓ ❑ Το πεδίο της γνώσης του προβλήματος είναι περιορισμένο 	<ul style="list-style-type: none"> ❖ Πολλές φορές τα προβλήματα απαιτούν πολλαπλούς τομείς γνώσης για να επιλυθούν ❑ Ένας ειδικός δεν μπορεί να έχει εμπειρία σε όλα
<ul style="list-style-type: none"> ❖ Διευκολύνεται ο χρονοπρογραμματισμός των συναντήσεων 	<ul style="list-style-type: none"> ❖ Όταν υπάρχει ένας μόνο ειδικός είναι πιο πιθανό να αναβάλλει τις συναντήσεις, παρά όταν είναι πολλοί
<ul style="list-style-type: none"> ❖ Οι ασυνέπειες και οι αντιφάσεις στη γνώση είναι ευκολότερο να αντιμετωπιστούν 	<ul style="list-style-type: none"> ❖ Ένας μόνο ειδικός → μία μοναδική συλλογιστική οδός ❑ Αυθεντία: Η γνώμη του ειδικού ακολουθείται «τυφλά» ❑ Το πεδίο της γνώσης δεν μπορεί να εξεταστεί σε βάθος

❖ Με λίγα άτομα παρόντα, ο ειδικός «ανοίγεται» πιο εύκολα για να δώσει τη γνώση του

❖ Αν ο ειδικός δεν έχει επικοινωνιακές ικανότητες τότε η απόκτηση της γνώσης δυσχεραίνει

Χρήση πολλών Ειδικών

Πλεονεκτήματα	Μειονεκτήματα
❖ Τα πολύπλοκα προβλήματα απαιτούν πολλές ειδικότητες για να λυθούν	❖ Δυσκολίες συντονισμού ❖ Πολλές φορές απαιτείται να υπάρχουν περισσότεροι μηχανικοί γνώσης
❖ Όσο πιο πολλές και διαφορετικές γνώμες ακούει ο μηχανικός της γνώσης, τόσο πιο πολύ μπορεί να εμβαθύνει στο πεδίο της γνώσης γιατί μπορεί να το «δει» από πολλές σκοπιές	❖ Ο μηχανικός γνώσης μπορεί να αποπροσανατολιστεί από τις διαφορετικές απόψεις
❖ Η συνεργασία και ο διάλογος πολλών ειδικών μπορεί να δημιουργήσει γνώση	❖ Συχνά υπάρχουν διαφωνίες
❖ Οι επίσημες συναντήσεις είναι πολλές φορές πρόκληση για προσφορά και	❖ Θέματα εμπιστευτικότητας

δημιουργικότητα εκ μέρους των ειδικών (ανταγωνισμός)	
---	--

Μεθοδολογίες Εκμαίευσης Γνώσης

Συνέντευξη

- ❖ Ο πιο διαδεδομένος και αποδοτικός τρόπος εκμαίευσης γνώσης
- ❑ Χρησιμοποιούνται συνήθως στα αρχικά στάδια απόκτησης της γνώσης
- ❑ Απαιτεί εκτεταμένη προετοιμασία και εξάσκηση από την πλευρά του μηχανικού
- ❑ Είναι χρονοβόρα διαδικασία
- ❖ **Πλεονέκτημα:** άμεση επαφή με τον ειδικό και παρατήρηση της συμπεριφοράς του

Είδη Συνεντεύξεων

- ❖ **Μη-δομημένες** συνεντεύξεις.
- ❑ Γενικές ερωτήσεις που υποβάλλονται με την ελπίδα της καταγραφής όσο περισσότερων πληροφοριών
- ❖ **Ημιδομημένες** συνεντεύξεις.
- ❑ Σειρά ανοιχτών ερωτήσεων και θεμάτων που πρέπει να καλυφθούν.
- ❖ **Δομημένες** συνεντεύξεις.
- ❑ Ερωτηματολόγιο με αυστηρά καθορισμένη δομή που περιλαμβάνει συγκεκριμένες ερωτήσεις σχετικές με τα χαρακτηριστικά του προβλήματος

Τεχνικές Συνέντευξης (1/4)

❖ Άμεση Παρατήρηση

- Ο μηχανικός γνώσης πρέπει να παρατηρεί, να ερμηνεύει και να καταγράφει τις διαδικασίες επίλυσης προβλημάτων στο φυσικό τους χώρο
- Η παρατήρηση μπορεί να εκτρέψει την προσοχή των υπολοίπων υπαλλήλων
- Η καταγραφή των γεγονότων απέχει χρονικά από τα ίδια τα γεγονότα, συνεπώς μπορεί να υπάρξουν λάθη
- Είναι απαραίτητο οι επισκέψεις να είναι σύντομες και επαναλαμβανόμενες

❖ Ανάλυση Πρωτοκόλλου

- Ο ειδικός καλείται να λύσει ένα πρόβλημα σκεπτόμενος μεγαλόφωνα
- Αποτελεσματικός τρόπος καταγραφής της διαδικασίας σκέψης
- Βοηθάει τον ειδικό να συνειδητοποιήσει τις διαδικασίες τις οποίες περιγράφει
- Η μέθοδος βοηθάει αργότερα στην αναπαράσταση της γνώσης

Τεχνικές Συνέντευξης (2/4)

❖ Επαναδιδασκαλία (teach-back):

- ❑ Ο μηχανικός γνώσης προσπαθεί να επαναδημιουργήσει και να συνοψίσει ότι έχει ειπωθεί από τον ειδικό και να το διδάξει σε αυτόν.
- ❑ *Μειονέκτημα:* Ο ειδικός μπορεί να κάνει το λάθος να εγκρίνει την επαναδιδασκαλία, χωρίς να εμβαθύνει σε έλεγχο για πληρότητα και η συνέπεια

❖ Διδακτική συνέντευξη (tutorial interview):

- ❑ Ο ειδικός δίνει μια διάλεξη πάνω στην περιοχή του θέματος.
- ❑ *Μειονέκτημα:* Δεν επιτρέπει στο μηχανικό γνώσης να επιβάλλει τη δομή της διαδικασίας

❖ Ταξινόμηση καρτών (card sorting)

- ❑ Ανακάλυψη κατάλληλων ιδιοτήτων των στοιχείων που απαρτίζουν την περιοχή του πεδίου, για την ταξινόμηση των εννοιών.
- ❑ Κάθε στοιχείο της περιοχής γράφεται σε μία κάρτα.
- ❑ Ο μηχανικός γνώσης ζητάει από τον ειδικό να ταξινομήσει τις κάρτες σε σωρούς.
- ❑ Ρωτάει τη βάση στην οποία στηρίχτηκε η ταξινόμηση
- ❑ Ρωτάει παραπλήσιες ερωτήσεις για κάθε στοιχείο μέλος του σωρού.
- ❑ *Ιεραρχική ταξινόμηση:* Μετά την πρώτη ταξινόμηση, κάθε σωρός θεωρείται σαν ένας χώρος (domain) και μπορεί να ταξινομηθεί ξεχωριστά σε υποσωρούς

Τεχνικές Συνέντευξης (3/4)

Βαθμωτά πλέγματα (laddered grids)

- ❖ Οι έννοιες θεωρείται ότι ανήκουν σε έναν άξονα γενικού-ειδικού
- ❖ Γίνονται ερωτήσεις:
 - ❑ Στον άξονα γενικού-ειδικού, ή αντίθετα
 - ❑ Κάθετα στον άξονα
- ❖ Εκκίνηση από έννοια-"σπόρο"
 - ❑ Ανάπτυξη δικτύου περιγραφής των σχέσεων των στοιχείων του θέματος (domain items)
- ❖ Κατευθυντικές ερωτήσεις:
 - ❑ Για τη μετακίνηση από το γενικό προς το ειδικό:
 - "Μπορείς να δώσεις παράδειγμα του ..."
 - "Πώς μπορείς να εξηγήσεις ότι ..."
 - ❑ Για τη μετακίνηση από το ειδικό προς το γενικό:
 - "Τι κοινό υπάρχει ..."
 - "Τι παραδείγματα υπάρχουν από ..."
 - "Τι διαφορές υπάρχουν από ..."
 - ❑ Για την κάθετη μετακίνηση:
 - "Τι εναλλακτικά παραδείγματα του ... υπάρχουν ..."

Τεχνικές Συνέντευξης (4/4)

Πλέγματα Ρεπερτορίων (repertory grid)

- ❖ Ο ειδικός καλείται να κατηγοριοποιήσει το πεδίο του προβλήματος χρησιμοποιώντας το προσωπικό του μοντέλο
 - ❑ Το πλέγμα χρησιμεύει στη σύλληψη και στην αξιολόγηση του μοντέλου του ειδικού
- ❖ Το πλέγμα είναι ένας πίνακας διπλής εισόδου
 - ❑ Τα στοιχεία τοποθετούνται με διαβάθμιση
- ❖ Κάθε **στοιχείο της περιοχής** κατηγοριοποιείται σύμφωνα με ένα σύνολο από **έννοιες ή χαρακτηρισμούς**
 - ❑ Εφαρμόζονται σε όλα τα στοιχεία σε κάποιο βαθμό
- ❖ Κάθε έννοια εκφράζεται σε μια γραμμική, αριθμητική κλίμακα.
 - ❑ Η κλίμακα είναι ίδια κάθε φορά.
 - ❑ Τυπικά οι τιμές κυμαίνονται 1-5 ή 1-10
 - ❑ Υπάρχουν δύο ακραίες τιμές, π.χ. βαρύς/ελαφρύς, φτηνός/ακριβός, κ.α.
 - ❑ Η μέση τιμή (π.χ. 3 στα 5) αντιπροσωπεύει μια ενδιάμεση τιμή της έννοιας.
- ❖ Ζητείται από τον ειδικό να αποδώσει μια τιμή σε κάθε έννοια για όλα τα στοιχεία της περιοχής, στο **πλέγμα** που δημιουργείται.

Παράδειγμα Πλέγματος Ρεπερτορίων

	μικροκλοπή	διάρρηξη	ναρκωτικά	δολοφονία	ληστεία	βιασμός	
οποιοδήποτε	2	1	1	1	1	5	μόνο γυναίκες
μεγάλη καταδίκη	2	1	1	2	3	5	μικρή καταδίκη
ειδική τοποθεσία	2	5	1	1	4	5	οποιαδήποτε τοποθεσία
προσχεδιασμένο	5	3	1	2	5	4	αφθόρμητα
μη-απειλητικός	3	2	2	5	5	5	απειλητικός
απρόσωπο	2	2	1	5	4	5	προσωπικό
ασήμαντο	1	3	1	5	4	5	σημαντικό
μη-βίαιος	1	1	2	5	5	5	βίαιος

μικροκλοπή
 διάρρηξη
 ναρκωτικά
 δολοφονία
 ληστεία
 βιασμός

μικροκλοπή	διάρρηξη	ναρκωτικά	δολοφονία	ληστεία	βιασμός
-	10	10	18	15	23
10	-	10	18	15	21
10	10	-	16	21	29
18	18	16	-	9	13
15	15	21	9	-	10
23	21	29	13	10	-

Επεξεργασία Πλεγμάτων Ρεπερτορίων

- ❖ Εξετάζεται αν κάποιο ζευγάρι είναι παρόμοιο κατά τη σύγκριση των οριζοντίων γραμμών του πλέγματος
 - Παραλείπονται κάποιες παραπλήσιες έννοιες
- ❖ Υπολογίζεται, σε ένα νέο πλέγμα, πόσο όμοια ή ανόμοια είναι τα στοιχεία της περιοχής μεταξύ τους.
- ❖ *Πλεονέκτημα:* Αναγκάζει τον ειδικό να σκεφτεί το πρόβλημα πιο σοβαρά
- ❖ *Μειονέκτημα:* Είναι δύσκολη η διαχείριση των μεγάλων πλεγμάτων
- ❖ Η μέθοδος χρησιμοποιείται συνήθως στα αρχικά στάδια της απόκτησης της γνώσης

Επαλήθευση (verification)

Η ορθή ανάπτυξη του συστήματος (O'Keefe, 1987)

Building the system right.

- ❖ Έλεγχος της συμβατότητας του συστήματος με τις αρχικές προδιαγραφές.
- ❖ Επιβεβαίωση της συνέπειας και πληρότητας της κωδικοποίησης της γνώσης που περιέχεται στο έμπειρο σύστημα.
 - ❑ Έλεγχος λαθών που οφείλονται στους κατασκευαστές του συστήματος.
 - ❑ Δεν ελέγχεται η ίδια η γνώση που εκμαιεύτηκε αλλά ο τρόπος με τον οποίο υλοποιήθηκε.
 - ❑ Ο έλεγχος πραγματοποιείται από το μηχανικό της γνώσης με τη βοήθεια εργαλείων (π.χ. CHECK, TEIRESIAS)
- ❖ Υπάρχουν λάθη
 - ❑ Σε συστήματα Κανόνων
 - ❑ Στις δομημένες αναπαραστάσεις γνώσης.

Λάθη στις Δομημένες Αναπαραστάσεις Γνώσης

- ❖ Οφείλονται συνήθως σε εννοιολογικές παρανοήσεις
- ❖ *Λάθη ταξινόμησης*
 - ❑ Κάποια κλάση τοποθετείται σε άλλο σημείο της ιεραρχίας από εκείνο που θα έπρεπε.
- ❖ *Λάθη ιδιοτήτων*
 - ❑ Κάποιες ιδιότητες (slots) τοποθετούνται σε λάθος σημεία της ιεραρχίας.
 - ❑ Π.χ. αν κάποια ιδιότητα s αποδίδεται και στις δύο υποκλάσεις B και C της κλάσης A , τότε το ορθότερο είναι η ιδιότητα αυτή να ορισθεί στην κλάση A αντί στις υποκλάσεις της.
- ❖ *Λάθη τιμών*
 - ❑ Οι ιδιότητες κάποιων αντικειμένων είτε παίρνουν τιμές αντίθετες με τους περιορισμούς της ιδιότητας, είτε παίρνουν την εξ' ορισμού τιμή, ενώ δεν είναι απαραίτητο.

Λάθη στα Συστήματα Κανόνων

- ❖ Πιθανά συντακτικά και σημασιολογικά λάθη σε συστήματα κανόνων.
- ❖ Επηρεάζουν την ορθότητα του συστήματος
 - ❑ Πλεονάζοντες κανόνες (redundant rules)
 - ❑ Αντικρουόμενοι κανόνες (conflicting rules)
 - ❑ Υπονοούμενοι κανόνες (subsumed rules)
 - ❑ Κυκλικοί κανόνες (circular rules)
 - ❑ Περιττές συνθήκες (unnecessary conditions)
- ❖ Επηρεάζουν την πληρότητα του συστήματος
 - ❑ Αδιέξοδοι κανόνες (dead-end rules)
 - ❑ Απόντες κανόνες (missing rules)
 - ❑ Απρόσιτοι κανόνες (unreachable rules)

Πλεονάζοντες Κανόνες (redundant rules)

❖ *Συντακτικός πλεονασμός*: Ίδιες συνθήκες και συμπέρασμα.

```
(defrule rule1
  (humidity high)
  (temperature hot)
```

=>

```
(assert
  (weather thunderstorms)))
```

```
(defrule rule2
  (temperature hot)
  (humidity high)
```

=>

```
(weather (assert
  (weather thunderstorms)))
```

□ Μπορεί να προκαλέσουν προβλήματα όταν συνοδεύονται από συντελεστές βεβαιότητας, γιατί αυξάνουν τεχνητά τη βεβαιότητα του συμπεράσματος.

❖ *Σημασιολογικός πλεονασμός*: Συνθήκες ή/και συμπεράσματα μπορεί να είναι διαφορετικά στη σύνταξη αλλά ίδια στη σημασία.

```
(defrule rule3
  (humidity high)
  (temperature hot)
```

=>

```
(assert
  (weather thunderstorms)))
```

```
(defrule rule4
  (temperature hot)
  (humidity high)
```

=>

```
(weather (assert
  (weather electrical-
  storms)))
```

□ Είναι πιο σπάνιο φαινόμενο

□ Αντιμετωπίζεται δυσκολότερα λόγω αδυναμίας αυτόματου ελέγχου από το σύστημα της ομοιότητας των εννοιών

- Οφείλεται στη μη σωστή δόμηση του συστήματος εννοιών (πλαίσια)

Αντικρουόμενοι Κανόνες (conflicting rules)

❖ Ίδιες συνθήκες, διαφορετικά συμπεράσματα.

```
(defrule rule5  
  (temperature hot)  
  (humidity high)
```

=>

```
(assert  
  thunderstorms))
```

```
(defrule rule6  
  (temperature hot)  
  (humidity high)
```

=>

```
(weather (assert (weather sunshine)))
```

Υπονοούμενοι Κανόνες (subsumed rules)

- ❖ Αν ένας κανόνας έχει περισσότερους περιορισμούς στη συνθήκη του από έναν άλλο, ενώ και οι δύο έχουν το ίδιο συμπέρασμα.

```
(defrule rule7
  (temperature hot)
  (humidity high)
  (pressure low)
=>
  (assert
    (weather thunderstorms)))

(defrule rule8
  (temperature hot)
  (humidity high)
=>
  (assert
    (weather thunderstorms)))
```

- ❖ Μόνο ένας κανόνας είναι απαραίτητος.

□ Είτε ο **rule7** είναι πολύ εξειδικευμένος ενώ δε χρειάζεται, ή ο **rule8** είναι πολύ γενικός.

- ❖ Αν υπάρχουν συντελεστές βεβαιότητας, τότε μπορεί οι υπονοούμενοι κανόνες να χρησιμεύουν στην αύξηση της βεβαιότητας του συμπεράσματος.

```
(defrule rule7
  CF=0,5
  ...)

(defrule rule8
  CF=0,7
  ...)
```

- Αν ισχύει μόνο **temperature hot** και **humidity high** η βεβαιότητα καταιγίδας είναι 0,7 (λόγω **rule8**).

□ Αν επιπλέον ισχύει ότι **pressure low**, τότε η βεβαιότητα αυξάνει σε 0,85 (λόγω **rule8** και **rule7**).

Κυκλικοί Κανόνες (circular rules)

- ❖ Δημιουργούν προβλήματα τερματισμού (αέναιος βρόχος - infinite loop)
- ❖ Εμφανίζονται με 2 μορφές:

□ *Συντακτικά κυκλικοί κανόνες:* Η συνθήκη κάποιου κανόνα αποτελεί συμπέρασμα κάποιου άλλου και αντίστροφα.

```
(defrule rule9  
  (brothers ?x ?y)  
=>  
  (assert (same-parents ?x ?y)))  
  
(defrule rule10  
  (same-parents ?x ?y)  
=>  
  (assert (brothers ?x ?y)))
```

- Δημιουργείται πρόβλημα μόνο στα συστήματα παραγωγής που δεν ελέγχουν αν κάποιο συμπέρασμα έχει εισαχθεί ξανά.

□ *Κυκλικά δεδομένα:* Η συνθήκη και το συμπέρασμα κάποιου κανόνα αναφέρονται σε δεδομένα που συνδέονται μεταξύ τους κυκλικά.

```
(defrule rule11  
  (important-city ?x)  
  (connected-with-road ?x ?y)  
=>  
  (assert (important-city ?y)))
```

- Δημιουργείται πρόβλημα τερματισμού μόνο αν τα δεδομένα έχουν κυκλική αλληλεξάρτηση.

Περισσότερες Συνθήκες (unnecessary conditions)

- ❖ Δύο κανόνες με ίδια συμπεράσματα αλλά περίπου ίδιες συνθήκες

```
(defrule rule12
  (patient ?x)
  (has-pink-spots ?x)
  (has-fever ?x)
```

=>

```
(assert (has-measles ?x))
```

```
(defrule rule12
```

```
(patient ?x)
  (has-pink-spots ?x)
  (not (has-fever ?x))
```

=>

```
(assert (has-measles ?x))
```

- ❖ Η συνθήκη **has-fever** φαίνεται μη αναγκαία για να έχει κάποιος ιλαρά.

- Οι 2 κανόνες πρέπει να συμπτυχθούν σε 1:

```
(defrule rule14
```

```
(patient ?x)
  (has-pink-spots ?x)
```

=>

```
(assert (has-measles ?x))
```

- ❖ Πολλές φορές τέτοιες καταστάσεις προκαλούνται όχι λόγω άχρηστων συνθηκών αλλά λόγω:

- Ελλιπών συνθηκών
- Λάθος κωδικοποίηση της γνώσης
- Λάθος στην εκμειευμένη γνώση

Αδιέξοδοι Κανόνες (dead-end rules)

❖ Κανόνες με συμπεράσματα τα οποία:

- ❑ Δεν ανήκουν στα τελικά συμπεράσματα του συστήματος.
- ❑ Δεν εμφανίζονται στις συνθήκες άλλων κανόνων.

❖ Παράδειγμα

```
(defrule rule15
```

```
  (gas-gauge empty)
```

```
=>
```

```
  (assert (gas-tank empty)))
```

❑ Είναι αδιέξοδος κανόνας, αν το συμπέρασμα **gas-tank empty**:

- Δεν ανήκει στους τελικούς στόχους του συστήματος.
- Δεν εμφανίζεται στη συνθήκη κάποιου άλλου κανόνα.

❖ Προβλήματα που προκαλούν οι αδιέξοδοι κανόνες:

- ❑ Άχρηστα συμπεράσματα (πρόβλημα απόδοσης του συστήματος)
- ❑ Ένδειξη για κανόνες που λείπουν (πρόβλημα ορθότητας-πληρότητας του συστήματος)

Απόντες Κανόνες (missing rules)

- ❖ Η απουσία κανόνων μπορεί να γίνει αντιληπτή από:
 - ❑ Παρουσία γεγονότων που δεν εμφανίζονται στη συνθήκη κανενός κανόνα.
 - ❑ Παρουσία τελικών συμπερασμάτων που δεν εμφανίζονται στο συμπέρασμα κανενός κανόνα.
 - ❑ Ύπαρξη αδιέξοδων κανόνων.

Απρόσιτοι Κανόνες (unreachable rules)

- ❖ Κανόνες που δεν ενεργοποιούνται ποτέ γιατί οι συνθήκες στις οποίες στηρίζονται δεν αποτελούν το συμπέρασμα κανενός κανόνα, ούτε ανήκουν στα αρχικά δεδομένα.
- ❖ Είναι ακριβώς το αντίθετο των αδιέξοδων κανόνων.

Έλεγχος Αξιοπιστίας (validation)

Η ανάπτυξη του σωστού συστήματος (O'Keefe, 1987)

Building the right system.

- ❖ Διαπίστωση της ορθότητας του τελικού συστήματος σε σχέση με τις ανάγκες και απαιτήσεις του τελικού χρήστη.
 - ❑ Επιβεβαίωση της ορθότητας των αποτελεσμάτων του συστήματος
 - ❑ Επιβεβαίωση ότι το σύστημα ανταποκρίνεται στις απαιτήσεις των χρηστών
- ❖ Τελικός ποιοτικός έλεγχος στην ανάπτυξη συστημάτων γνώσης.
 - ❑ Ελέγχεται η ορθότητα και γενικότητα της ίδιας της γνώσης που εκμαιεύτηκε και περιέχεται στο έμπειρο σύστημα
 - ❑ Ελέγχεται αν το σύστημα επιλύει τα προβλήματα με ορθό και επακριβή τρόπο

Μεθοδολογίες Ελέγχου Αξιοπιστίας

- ❖ *Άτυπος έλεγχος*: Συναντήσεις μηχανικού γνώσης με ειδικούς κατά τη διάρκεια ανάπτυξης του συστήματος για τον έλεγχο της εγκυρότητας κάποιων αποτελεσμάτων
- ❖ *Δοκιμασία σε υποδειγματικές περιπτώσεις (test cases)*
 - ❑ Οι λύσεις που δίνει το έμπειρο σύστημα συγκρίνονται με λύσεις που δόθηκαν από διάφορους ειδικούς του τομέα στις ίδιες περιπτώσεις.
 - ❑ Οι ειδικοί συμφωνούν ή διαφωνούν (ίσως διαβαθμισμένα) με τις υποδείξεις του έμπειρου συστήματος.
 - ❑ *Δοκιμασία Turing*: Οι λύσεις ειδικών και συστήματος παρουσιάζονται με την ίδια μορφή σε άλλους ειδικούς, οι οποίοι τις αξιολογούν αντικειμενικά.
 - ❑ Οι υποδειγματικές περιπτώσεις πρέπει να είναι διαφορετικές από αυτές που χρησιμοποιήθηκαν στις προηγούμενες φάσεις ανάπτυξης του συστήματος.
 - Εξασφαλίζεται η **ευρωστία (robustness)** σε μη-προσδοκώμενα δεδομένα.
 - ❑ Όταν αυξάνεται η πολυπλοκότητα του συστήματος, ο αριθμός των δοκιμασιών που θα έπρεπε να πραγματοποιηθούν αυξάνεται εκθετικά.
- ❖ *Δοκιμασία σε πραγματικές συνθήκες*
 - ❑ Κίνδυνος απώλειας εμπιστοσύνης από τους τελικούς χρήστες
 - ❑ Πρέπει να γίνεται κοντά στο τελικό στάδιο ανάπτυξης

Μεθοδολογίες Ελέγχου Αξιοπιστίας

❖ *Έλεγχος αξιοπιστίας των υποσυστημάτων:* Το σύστημα γνώσης χωρίζεται σε ανεξάρτητα υποσυστήματα τα οποία ελέγχονται ξεχωριστά

❑ Ευκολότερη η επίλυση προβλημάτων σε μικρότερα συστήματα

❑ Η αξιοπιστία του καθενός υποσυστήματος ξεχωριστά δεν εγγυάται πάντα την αξιοπιστία του συνολικού συστήματος

❖ *Ανάλυση ευαισθησίας:* Δοκιμασία με σύνολο παραμέτρων που διαφέρουν λίγο σε μία από τις παραμέτρους κάθε φορά

❑ Ιδιαίτερα χρήσιμος έλεγχος σε συστήματα με αβεβαιότητα.

Κριτήρια Αξιοπιστίας

❖ Σύγκριση με γνωστά αποτελέσματα

❖ Σύγκριση με την απόδοση ειδικών

❑ Μεγαλύτερη ανεκτικότητα σε λάθη, αφού και ο ειδικός μπορεί να κάνει λάθη

❖ Σύγκριση με αποτελέσματα που προβλέπονται θεωρητικά

❑ Συνήθως γίνεται όταν το σύστημα γνώσης μοντελοποιεί κάποια φυσική διαδικασία

❑ Όταν δεν υπάρχει ακριβές θεωρητικό μοντέλο, γιατί το φυσικό σύστημα είναι πολύπλοκο, τότε δεν είναι δυνατή αυτή η σύγκριση

Μέτρηση Αξιοπιστίας

❖ *Ακρίβεια (accuracy)*: Ποσοστό των αποδεκτών απαντήσεων του συστήματος

□ Αποδεκτές απαντήσεις είναι αυτές που συμπίπτουν σε αυτές ενός ειδικού

❖ *Επάρκεια (adequacy)*: Ποσοστό κάλυψης (*coverage*) του πεδίου γνώσης του προβλήματος

□ Π.χ. ένα σύστημα κατηγοριοποίησης αναγνωρίζει σωστά το 83% των ειδών

□ Θα μπορούσε το ποσοστό να περιέχει και βάρη, δίνοντας μεγαλύτερη έμφαση στα σημαντικότερα στοιχεία του πεδίου της γνώσης

Λάθη στην Αναπαράσταση της Γνώσης

❖ *Λάθη απόφασης*: Συμβαίνουν όταν το σύστημα καταλήγει σε λάθος αποτέλεσμα

□ Επηρεάζουν την ακρίβεια του συστήματος

□ Διαπιστώνονται εύκολα, αλλά εντοπίζονται και διορθώνονται δύσκολα

❖ *Λάθη παράλειψης*: Συμβαίνουν όταν το σύστημα δεν μπορεί να καταλήξει σε αποτέλεσμα

□ Η απαραίτητη γνώση για να λυθεί κάποιο πρόβλημα παραλήφθηκε

□ Επηρεάζουν την επάρκεια του συστήματος

□ Διαπιστώνονται δύσκολα γιατί η δοκιμαστική περίπτωση (*test case*) που θα αποκαλύψει την έλλειψη δεν είναι προφανής στο μηχανικό γνώσης