

Τεχνητή Νοημοσύνη Artificial Intelligence

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 1

Τεχνητή Νοημοσύνη (Artificial Intelligence)

- ΤΙ ΕΙΝΑΙ ΝΟΗΜΟΣΥΝΗ;
 - Η ικανότητα κρίσης, σκέψης, συμπερασμού
 - Η ικανότητα απόκτησης / συλλογής και χρήσης γνώσεων
 - Η ικανότητα αντίληψης και συνδιαλλαγής με αντικείμενα του έξω κόσμου
 - Ένα αμάλγαμα πολλών δραστηριοτήτων που είναι σχετικές με την αναπαράσταση και επεξεργασία πληροφοριών

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 2

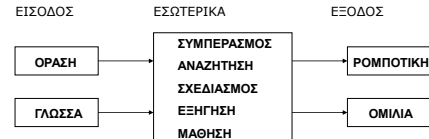
ΤΙ ΕΙΝΑΙ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

- Η μελέτη για το πώς ένας υπολογιστής να μπορεί να κάνει πράγματα στα οποία, μέχρι σήμερα, οι άνθρωποι είναι καλύτεροι
- Η ανάλυση των αρχών που διέπουν την λειτουργία ενός νοήμονα επεξεργαστή πληροφοριών
- Η μελέτη στοιχείων και ιδεών που μπορούν να κάνουν ένα υπολογιστή να έχει νοήμονα συμπεριφορά
- Η ανάπτυξη τεχνικών και μεθόδων για την επίλυση πολύπλοκων προβλημάτων που βασίζονται στη χρήση γνώσης γύρω από την συγκεκριμένη περιοχή

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 3

ΤΙ ΕΙΝΑΙ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

- ΤΝ είναι η μελέτη νοητικών λειτουργιών με τη χρήση υπολογιστικών μοντέλων
- Κεντρικό δόγμα της ΤΝ:
- Ότι κάνει το μαλάλο μπορεί να θεωρηθεί ότι είναι, σε κάποιο επίπεδο, ένα είδος υπολογισμού
- Νοητικές λειτουργίες



ΠΟΙΑ ΕΙΝΑΙ Η ΦΥΣΗ ΑΥΤΟΥ ΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΕΙΝΑΙ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ Τ.Ν.

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 4

ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΙΝΑΙ ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ Τ.Ν.

- ΠΑΙΓΝΙΔΙΑ
- ΑΠΟΔΕΙΞΗ ΘΕΩΡΗΜΑΤΩΝ
- ΓΕΝΙΚΟΣ ΕΠΙΛΥΤΗΣ ΠΡΟΒΛΗΜΑΤΩΝ
- ΑΝΤΙΛΗΨΗ
 - ΟΡΑΣΗ
 - ΟΜΙΛΙΑ
- ΚΑΤΑΝΟΗΣΗ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ
- ΕΜΠΕΙΡΗ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ
 - Συμβολικά μαθηματικά
 - Ιατρική διάγνωση
 - Χημική ανάλυση
 - Μηχανική σχεδίαση
- ΡΟΜΠΟΤΙΚΗ
- ΕΡΩΤΗΜΑΤΑ
 - Ποιες είναι οι βασικές υποθέσεις περί νοημοσύνης
 - Ποια είδη τεχνικών είναι χρήσιμα για επίλυση προβλημάτων Τ.Ν.
 - Σε ποιο επίπεδο λεπτομέρειας προσπαθούμε να μοντελοποιήσουμε την ανθρώπινη νοημοσύνη
 - Πώς θα ξέρουμε ότι έχουμε πετύχει να φτιάξουμε ένα νοήμων πρόγραμμα

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 5

Βασική Υπόθεση

- **ΕΡΩΤΗΜΑΤΑ**
 - Ποιες είναι οι βασικές υποθέσεις περί νοημοσύνης
 - Ποια είδη τεχνικών είναι χρήσιμα για επίλυση προβλημάτων Τ.Ν.
 - Σε ποιο επίπεδο λεπτομέρειας προσπαθούμε να μοντελοποιήσουμε την ανθρώπινη νοημοσύνη
 - Πώς θα ξέρουμε ότι έχουμε πετύχει να φτιάξουμε ένα νοήμων πρόγραμμα
- **Η ΒΑΣΙΚΗ ΥΠΟΘΕΣΗ**
 - Η υπόθεση του Newell και Simon (1976)
 - Η υπόθεση του ΦΥΣΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΥΜΒΟΛΩΝ (ΦΣΣ)
 - "Ένα ΦΣΣ έχει τα ικανά και αναγκαία μέσα για γενική νοήμονα δράση"

Κ. Χαλάντζης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 6

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΤΙ ΕΙΝΑΙ ΜΙΑ ΤΕΧΝΙΚΗ ΤΗΣ Τ.Ν.

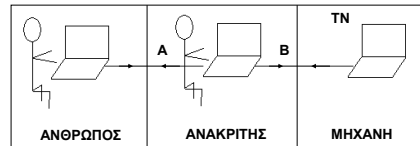
- **ΓΕΓΟΝΟΣ:** Η νοημοσύνη απαιτεί γνώση πού είναι πολυπληθής
 - δύσκολα χαρακτηρίζεται με ακρίβεια
 - αλλάζει συνεχώς
- **ΣΥΜΠΕΡΑΣΜΑ:** Μια τεχνική ΤΝs είναι μια μέθοδος που εκμεταλλεύεται ΓΝΩΣΗ που παριστάνεται με τρόπο που:
 - συλλαμβάνει γενικότητες
 - είναι αντιληπτή από ανθρώπους που πρέπει να την παράσχουν
 - μεταβάλλεται εύκολα (διόρθωση, ενημέρωση)
 - χρησιμοποιείται σε πολλές περιπτώσεις ακόμα και αν είναι ατελής και όχι απόλυτα σωστή
 - χρησιμοποιεί στο να ξεπεράσουμε το πολυπληθές της στενεύοντας την περιοχή αναζήτησης

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 7

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΚΡΙΤΗΡΙΟ ΕΠΙΤΥΧΙΑΣ

- **Η δοκιμή Turing:** Ο ανακριτής συνομιλεί με τον άνθρωπο και την μηχανή αλλά δεν ξέρει ποιος είναι ποιος



- Πρόβλημα: Ποιος είναι ο Α και ποιος ο Β
- Στόχος μηχανής: Να ξεγελάσει τον ανακριτή ώστε να νομίζει ότι μιλά σε άνθρωπο
- Σύνθετος κριτήριο: Σύγκριση με τον άνθρωπο
- Προγράμματα σκακιού
- DENDRAL: για την αναγνώριση οργανικών ενώσεων

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 8

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΕΙΔΗ ΣΥΛΛΟΓΙΣΜΟΥ (ΣΥΜΠΕΡΑΣΜΟΙ) 1/2

- **MODUS PONENS** (μια από τις 19 ταυτολογίες του Αριστοτέλη)

$$\begin{matrix} p \\ \Lambda p \Rightarrow q \end{matrix} \Rightarrow q$$
- **Καθολική εξειδίκευση (universal specialisation)**

$$(Vx)p(x) \Rightarrow p(A)$$
- **Παραγωγή (Deduction):** Κάθε λογικά σωστός συμπερασμός
- **Επαγωγή (Induction)**

$$P(A)\wedge P(B)\wedge \dots \Rightarrow (Vx)p(x)$$
- **Απαγωγή (Abduction) (όχι σωστός συμπερασμός)**

$$\begin{matrix} q \\ \Lambda p \Rightarrow q \end{matrix} \Rightarrow p$$

q: παρατήρηση
 $p \Rightarrow q$ έστω κανόνας
 p μια εξήγηση δυνατή αλλά όχι αναγκαία σωστή

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 9

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΕΙΔΗ ΣΥΛΛΟΓΙΣΜΟΥ (ΣΥΜΠΕΡΑΣΜΟΙ) 2/2

- Αναγωγή (Resolution - PROLOG + ενοποίηση)

$$\begin{matrix} p \\ \Lambda \neg p \vee q \end{matrix} \Rightarrow q$$

Εμπρόσθιος / οπίσθιος συλλογισμός (Forward/Backward reasoning
 $p \Rightarrow q$)

Εμπρόσθιος: από p προς το q
 Οπίσθιος: από q προς p
- Πιθανοκρατικός συλλογισμός / πιθανολογική γνώση (τα συμπεράσματα κανόνων ισχύουν με κάποια πιθανότητα μικρότερη της μονάδας
 $p \Rightarrow q$ με πιθανότητα P (0,1)
- Μη μονότονη συλλογιστική
 Συμπεράσματα μπορεί να ανασκευαστούν εν όψει νέας μαρτυρίας, σε αντίθεση με την μονότονη συλλογιστική στην οποία το σύνολο των συμπερασμάτων δεν μειώνεται.

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 10

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Κατανόηση φυσικής γλώσσας Το πρόγραμμα ELIZA (Turing Test του J. Weizenbaum, 1960) 1/2

Το πρόγραμμα ELIZA προσομοιώνει ένα Ψυχίατρο που συνομιλεί με μια ασθενή:

MEN ARE ALL ALIKE : ασθενής
 IN WHAT WAY : μηχανή - ψυχίατρος

THEY'RE ALWAYS BUGGING US ABOUT SOMETHING OR OTHER
 CAN YOU THINK OF A SPECIFIC EXAMPLE?

WELL, MY BOYFRIEND MADE ME COME HERE
 YOUR BOYFRIEND MADE YOU COME HERE

HE SAYS I'M DEPRESSED MOST OF THE TIME
 I'M SORRY TO HEAR YOU ARE DEPRESSED

IT'S TRUE, I AM UNHAPPY
 DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY?

I NEED SOME HELP, THAT MUCH SEEMS CERTAIN
 WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP?

PERHAPS I COULD LEARN TO GET ALONG WITH MY MOTHER
 TELL ME MORE ABOUT YOUR FAMILY

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 11

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Το πρόγραμμα ELIZA (Turing Test του J. Weizenbaum, 1960) 2/2

MY MOTHER TAKES CARE OF ME
 WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

MY FATHER
 YOUR FATHER?

YOU ARE LIKE MY FATHER IN SOME WAYS
 WHAT RESEMBLANCE DO YOU SEE?

YOU ARE NOT AGGRESSIVE, BUT I THINK YOU DON'T WANT ME TO NOTICE THAT
 WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE?

YOU DON'T ARGUE WITH ME
 THAT MAKES YOU THINK I DON'T ARGUE WITH YOU?

YOU ARE AFRAID OF ME
 DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU?

MY FATHER IS AFRAID OF EVERYBODY
 WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER?

BULLIES
 DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 12

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Τεχνικές του ELISA

> **Scripts του προγράμματος ELISA** = (templates ή skeletons)

T1: What makes you think XXX
T2: Tell me more about XXX
T3: Does it please you to believe XXX

Η πρόταση εισόδου εξετάζεται για την παρουσία **KEYWORDS**
 π.χ. I, You, mother, hate, ...

> Κάθε **KEYWORD** σχετίζεται με ένα σύνολο **κλώνων** που μετασχηματίζουν τη φράση εισόδου σε φράση εξόδου
 π.χ., **Rule: YYY you are ZZZ -> I am ZZZ**
 You are afraid of me -> I am afraid of you
 @ T3 -> Does it please you to believe I am afraid of you
 ή T1 -> What makes you think I am afraid of you

> Όταν το σύστημα δεν βρίσκει **KEYWORDS** τότε αντιδρά με φράση από προηγούμενη συζήτηση

> Όταν επίσης βλέπει να δίνονται μονολεκτικές απαντήσεις τότε αντιδρά με κάποια γενικής φύσης φράση όπως είναι:
YOU DON'T SEEM VERY TALKATIVE TODAY

> **ELISA ΔΕΝ ΚΑΤΑΝΟΕΙ ΤΗΝ ΦΥΣΙΚΗ ΓΛΩΣΣΑ**

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 13

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Προβλήματα στην κατανόηση της Φυσικής Γλώσσας

- ✓ Cinderella attended the ball
 - ✓ Cinderella kicked the ball
- ✓ The house was built by the workmen
 - ✓ The house was built by the river
- ✓ The soldiers shot the prisoners and I saw several fall
 - ✓ The soldiers shot the prisoners and I saw several reload
- ✓ Η Ελένη είδε την τανία με τον Δημήτρη
 - ✓ Η Ελένη είδε την τανία με τον Σαρλό
- ✓ Ο Γιώργος είδε τον Νίκο στην πλατεία με τα κιάλια
 - ✓ Ο Γιώργος είδε τον Νίκο στην πλατεία με το αγαλάμα
- ✓ Ο Γιώργος χτύπησε το σκύλο του
 - ✓ Ο Γιώργος χτύπησε το πόδι του
- ✓ Όλα τα βιβλία και τα περιοδικά που λείπουν κοστίζουν 200.000 δρχ.
 - ✓ Όλα τα βιβλία και τα περιοδικά που εκδίδονται ανά τρίμηνο κοστίζουν 200.000 δρχ.

> Η κατανόηση της φυσικής γλώσσας απαιτεί πολύ γνώση.
 > Απαιτούνται τέσσερα στάδια με αναδράσεις μεταξύ τους (για προτάσεις):

μορφολογική → συντακτική → σημασιολογική → πραγματική
 ανάληψη ανάληψη ανάληψη ανάληψη

> Για κατανόηση κειμένου απαιτείται επίσης και συζήτηση προτάσεων

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 14

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Μορφολογική ανάλυση

- Απομόνωση λέξεων
- Απομόνωση σημείων στίξης
- Μορφολογία λέξεων
 - Αναγνώριση μερών του λόγου
 - Διαχωρισμός θεμάτων από καταλήξεις
- Κλιτικό σύστημα
- Ελληνική γλώσσα
 - Σημαντική πολυπλοκότητα στην κλίση ρημάτων
 - Πολλές κλιτικές κατηγορίες (πλέον των 40) για ουσιαστικά και επίθετα

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 15

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Συντακτική Ανάλυση

- Στόχος
 - Συντακτικό δέντρο
- Κόμβοι συντακτικού δέντρου
 - Πρόταση/Π
 - Ονομαστική Φράση/ΟΦ
 - Ρηματική Φράση/ΡΦ
 - Εμπρόθετη Φράση/ΕΦ
 - Ουσιαστικό/Ο
 - Ρήμα/Ρ
 - Αρθρο/Α
 - ...
- Μέθοδοι
 - Δίετυα
 - Γραμματικές

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 16

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Παράδειγμα (με τη χρήση Augmented Transition Networks)

(Π (ΟΦ (Α (το))
 (Ο (παιδί)))
 (ΡΦ (Ρ (έγραψε))
 (ΟΦ (Α (τα))
 (Ο (μήλα)))))))

Π = Πρόταση
 ΟΦ = Ονομαστική Φράση
 Α = άρθρο
 Ο = όνομα
 ΡΦ = Ρηματική φράση
 Ρ = ρήμα

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 17

Πανεπιστήμιο Αθηνών Τμήμα Προφορικής και Τηλεπικοινωνιών

Σημασιολογική ανάλυση

- Στόχος
 - Σημασία
- Αναπαράσταση σημασίας
 - Μέθοδοι αναπαράστασης γνώσης
 - Συμβατή με το κεντρικό σύστημα
- Σημασία σε επίπεδο λέξεων
 - Λεκτική διασέφηση
- Σημασία σε επίπεδο προτάσεων
 - Σημασιολογικές γραμματικές
 - Πτωτικές γραμματικές
 - Ενωσιολογική εξάρτηση

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Προφορικής και των Τηλεπικοινωνιών 18

Κατανόηση κειμένου & Χειρισμός διαλόγων

- Τριτάκι αντισπέρμα
 - Ο Γιώργος είχε μια κόκκινη μάλλα. Ο Νίκος την ήθελε.
- Τρίψματα αντισπέρμα
 - Η Ελένη άνοιξε το βιβλίο που μόλις είχε αγοράσει. Η πρώτη σελίδα ήταν σκουισμένη.
- Τρίψματα ενεργειών
 - Ο Κώστας πήγε ταξίδι στην Κρήτη. Έφυγε με την πρωινή πτήση.
- Ουτότητα εμπλεκόμενων σε ενέργειες
 - Την περασμένη εβδομάδα έγινε διάρρηξη στο σπίτι μου. Μου έκλεψαν την τηλεόραση.
- Σχέσεις αιτίου-αιτιατού
 - Χθες είχε φοβερή χιονοθύελλα. Τα σχολεία είναι κλειστά σήμερα.
- Σωφές σχέδιων
 - Η Μαρία θέλει να αγοράσει καινούργιο αυτοκίνητο. Αποφάσισε να βρει δουλειά.

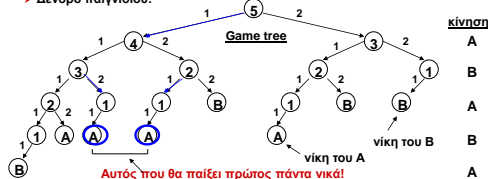
ΠΑΙΓΝΙΔΙΑ (GAMES)

- Είναι κοινά αποδεκτό μέσο μάθησης
- Ένα παιχνίδι επιδεικνύει πολλές πλευρές ευφυΐας
- Ειδικά στην περιοχή του σχεδιασμού (planning) τόσο στο άμεσο επίπεδο τακτικής, όσο και στο μακροπρόθεσμο επίπεδο στρατηγικής.
- Η τεχνητή νοημοσύνη έχει αναπτύξει διάφορες στρατηγικές στον τομέα των παιχνιδιών, όπως είναι:
 - το Δένδρο παιχνιδιού
 - ο Αλγόριθμος minimax

ΔΕΝΔΡΟ ΠΑΙΓΝΙΔΙΟΥ (game tree)

Π.χ. Το παιχνίδι NIM

- > σωρός 5 σπέρτων από τον οποίο δύο παίκτες (A και B) παίρνουν εναλλάξ 1 ή 2 σπέρτα. Αυτός που παίρνει τελευταίος χάνει!
- > Δένδρο παιχνιδιού:

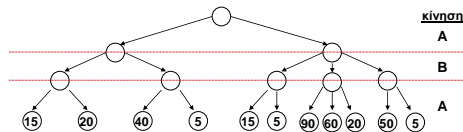


- > Δένδρο παιχνιδιού δυνατό μόνο για μικρά παιχνίδια: Σκεπτόμενος; evaluation function + look-ahead minimax search
- tic-tac-toe : 300.000 nodes
- checkers : 10⁴⁰ nodes
- chess : 10¹⁰⁰ nodes

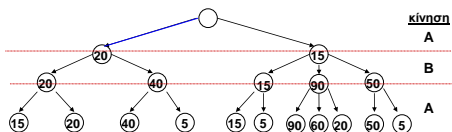
Συνάρτηση εκτίμησης & Πρόβλεψη (evaluation function + look ahead)

Minimax στρατηγική του A παίκτη:

- > Γέννηση του δένδρου παιχνιδιού για N κινήσεις μπροστά (N επίπεδα του δένδρου)
- > Υπολόγισε την τιμή της συνάρτησης εκτίμησης στα φύλλα του δένδρου
- > Εκκινώντας από τα φύλλα προς τη ρίζα, για κάθε επίπεδο του δένδρου ανέβασε σε κάθε κόμβο του είτε τη μέγιστη τιμή ή την ελάχιστη τιμή των κλάδων του κόμβου ανάλογα εάν οι κλάδοι του είναι κινήσεις του A ή αντίστοιχα του αντιπάλου του B.
- > Διάλεξη την κίνηση που οδηγεί σε κόμβο του 1ου επιπέδου με την μεγαλύτερη τιμή.



Συνάρτηση εκτίμησης & Πρόβλεψη (evaluation function + look ahead)



Διαδικασία minimax

```

module minimax(δένδρο, κίνηση)
{ όπου κίνηση: εγώ ή αντίπαλος}
if δένδρο = φύλλο {αυτό τερματίζει την αναδρομή}
then αποτέλεσμα = τιμή της συνάρτησης εκτίμησης
else if κίνηση = εγώ
then for κάθε κλάδο του δένδρου do
minimax(subtree, αντίπαλος)
αποτέλεσμα = maximum{των τιμών εκτίμησης}
else for κάθε κλάδο του δένδρου do
minimax(subtree, εγώ)
αποτέλεσμα = minimum{των τιμών εκτίμησης}
    
```

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Κλάδεμα του δένδρου β-κλάδεμα (β-cut)

> Δίνει τη δυνατότητα να μη γεννήσουμε όλο το δένδρο.
 > Π.χ., έστω ότι γεννά τους κόμβους από αριστερά-προς-τα-δεξιά και κατά-βάθος (δηλ., a, b, c, d, e, f, ...)

> Μετά την δημιουργία του κόμβου d έπεται ότι η τιμή της θέσης a ≤ 20
 > Μετά την δημιουργία του κόμβου f έπεται ότι η τιμή της θέσης e ≥ 40
 > Συνεπώς δεν υπάρχει λόγος να εξερευνηθώ άλλους κόμβους που ξεκινούν από τη θέση e (β-cut) δεδομένου ότι στον κόμβο a ανεβάζω τη minimum τιμή
 > Οπότε η τελική τιμή στον κόμβο a είναι ίση με a=20

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 25

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Κλάδεμα του δένδρου α-κλάδεμα (α-cut)

> Δίνει επίσης τη δυνατότητα να μη γεννήσουμε όλο το δένδρο.
 > Π.χ.,

• Μετά τη δημιουργία του κόμβου j έπεται ότι η τιμή της θέσης g ≤ 15
 • Επειδή όμως η τιμή της θέσης a είναι 20 έπεται ότι δεν χρειάζεται να εξερευνηθώ άλλους κόμβους που ξεκινούν από τον κόμβο g (α-cut)

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 26

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Προβλήματα

> Ο συνδυασμός των α-cut και β-cut περιορίζει σημαντικά τη γέννηση του δένδρου
 > Στο σχήμα του παραδείγματος φαίνεται η συνδυασμένη δράση των α και β cuts
 > Το α-β κλάδεμα δεν δίνει πάντα καλά αποτελέσματα σε θεμελιώδη δένδρα
 > Στην περίπτωση αυτή χρησιμοποιούνται ευρεστικά που δύσκολα ενσωματώνονται σε μια συνάρτηση εκτίμησης
 > Σημαντικό ρόλο παίζει η αναπαράσταση του προβλήματος, όπως θα δούμε με μερικά παραδείγματα.

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 27

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Το παιχνίδι Tic-Tac-Toe (τριλιζα)

□ Παίζουν εναλλάξ δύο παίκτες, ο X και ο O, που τοποθετούν το γράμμα τους σε μια θέση του 3x3 πίνακα (ο X παίζει πρώτος)
 □ Κερδίζει αυτός που πρώτος σχηματίζει οριζόντια, κάθετα ή διαγώνια μια τριάδα XXX, αντίστοιχα OOO

		X
	X	O
O		

□ Θα παρουσιάσουμε μια σειρά προγραμμάτων που παίζουν tic-tac-toe με αύξουσα

- πολυπλοκότητα
- χρήση γενικεύσεων
- καθαρότητα της γνώσης των
- επεκτασιμότητα

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 28

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

1ο πρόγραμμα Tic-Tac-Toe

α) Πίνακας: Διάγραμμα 9 στοιχείων
 π.χ., (0, 0, 2, 0, 1, 2, 1, 0, 0) 0: κενό
 1: X
 2: O

		O
	X	O
X		

β) Πίνακας κινήσεων: Διάγραμμα 3⁹ = 19.663 στοιχεία με κάθε στοιχείο = διάγραμμα πίνακα

γ) Αλγόριθμος:
 Για να κάνεις μια κίνηση
 1) Δες το διάγραμμα που παριστάνει τον πίνακα σαν ένα τριαδικό αριθμό και μετέτρεψέ τον σε δεκαδικό αριθμό
 2) Χρησιμοποίησε τον αριθμό σαν δείκτη στον πίνακα των κινήσεων και πάρε το φυλαγμένο εκεί διάγραμμα πίνακα
 3) Το διάγραμμα αυτό είναι η νέα μορφή του πίνακα, οπότε ενημέρωσε τον πίνακα

δ) Χαρακτηριστικά αλγορίθμου:
 • Ταχύτατος αλγόριθμος
 • Πολύ μνήμη για τον πίνακα κινήσεων
 • Απαιτεί προεργασία στη δημιουργία του πίνακα
 • Πρακτικά δεν επεκτείνεται (Οι 3 διαστάσεις απαιτούν 3²⁷ θέσεις)

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 29

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

2ο πρόγραμμα Tic-Tac-Toe

1/2

• Δομές δεδομένων

α) Πίνακας: όπως στο πρόγραμμα 1 αλλά

2: κενό	1	2	3
3: X	4	5	6
5: O	7	8	9

β) Αριθμοί κίνησης: 1,2,3,...,9

• Αλγόριθμος:
 procedures: make2 →

•	•	•
•	•	•
•	•	•

Posswin(p) = { 0 εάν ο παίκτης p δεν νικά
 else (νίκη = 3x3x2 ή 5x5x2)
 n η κίνηση νίκης

Go(n): κίνηση στο n τετράγωνο

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 30

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Κανάτες νερού

Αρχική κατάσταση (0, 0) Στόχος (2, 0)

Μια λύση:

1.	(X,Y X < 4) → (4,Y)	(0,0)	κανόνας
2.	(X,Y Y < 3) → (X,3)	(0,3)	2
3.	(X,Y X > 0) → (X-D,Y)	(3,0)	9
4.	(X,Y Y > 0) → (X,Y-D)	(3,3)	2
5.	(X,Y X > 0) → (0,Y)	(4,2)	7
6.	(X,Y Y > 0) → (X,0)	(0,2)	5
7.	(X,Y X+Y ≥ 4 & Y>0) → (4, Y-(4-X))	(2,0)	9
8.	(X,Y X+Y ≥ 3 & X>0) → (X-(3-Y), 3)		
9.	(X,Y X+Y ≥ 4 & Y>0) → (X+Y, 0)		
10.	(X,Y X+Y ≤ 3 & X>0) → (0, X+Y)		

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 37

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μηχανική Όραση

➢ Η Τεχνολογία στον τομέα αυτόν είναι ακόμη πρωτόγονη
 ➢ Μέθοδοι όρασης (για εικόνα 3D που απεικονίζονται στο 2D)

- Μετατροπή εικόνας σε ψηφιακή αναπαράσταση:
 - ψηφιοποίηση με σάρωση της εικόνας
 - πίνακας εντάσεων (intensity matrix)
- ανίχνευση έντονων περιοχών: γραμμές, γωνίες (συντακτική ανάλυση) μεταβολές έντασης
- συμπλήρωμα σχημάτων (σημασιολογική ανάλυση) γεωμετρικοί περιορισμοί
- Χαρακτηρισμός ακμών (edge labeling)
 - Αλγόριθμος φιλτραρίσματος (Waltz, 1972)
 - Huffman - Claves συμβολισμός
 - Επικέτες - επιγραφές (+, -, →, ←) ⇒ 4² και 4³ περιπτώσεις

➢ Ακόμη και η περίπτωση αναγνώρισης χειρογράφων είναι δύσκολη!

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 38

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πίνακας εντάσεων

Ο πίνακας αυτός δίνει την ένταση της δέσμης σάρωσης που ανακλάται πάνω στην εικόνα

Columns	4	5	6	7	8	
Rows	P	1	3	16	11	13
Q	2	3	15	15	14	
R	9	1	2	16	14	
S	7	2	2	3	16	

Ο πίνακας αυτός δίνει τις διαφορές εντάσεων ενόσ σημείου με τα τέσσερα γειτονικά του

Columns	4	5	6	7	8	
Rows	P	1	3	16	11	13
Q	1	1	1	1	1	1
R	1	1	1	1	1	1
S	1	1	1	1	1	1

Σημεία με παρόμοιες διαφορές συνδέονται και αντιπροσωπεύουν πιθανές ακμές

Figure 6.21 Processing an intensity matrix: (a) part of the matrix; (b) intensity gradients for the matrix of (a); (c) probable edges in the scene represented by (a)

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 39

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Συμπλήρωση γραμμών

Figure 6.22 Examples of processing incomplete line drawings: (a) joining collinear segments; (b) joining "hanging" lines

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 40

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Huffman - Claves συμβολισμός

Για χαρακτηρισμό των ακμών

+ κυρτή ακμή
 - κοίλη ακμή
 - ακμή με κρυμμένη τη μια επιφάνεια (η κατεύθυνση είναι τέτοια ώστε η κρυμμένη επιφάνεια είναι στα αριστερά)

φυσικά δυνατές γωνίες φυσικά αδύνατες γωνίες

• αναγνώριση αντικειμένων - πρότυπα σχήματα (πραγματική ανάλυση)

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 41

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Ερμηνεία της εικόνας

Μια ερμηνεία της εικόνας πρέπει να ικανοποιεί τους περιορισμούς:

α) κάθε γραμμή έχει την ίδια ετικέτα καθ' όλο το μήκος της } σημασιολογία
 β) οι διασταυρώσεις να είναι φυσικώς δυνατές

Οι μόνες δυνατές περιπτώσεις διασταυρώσεων

Κ. Χαλιώτης, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 42

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μη-διαδικαστικοί Αργόριθμοι 1/2

- Διαδικαστικός Ορισμός αλγορίθμων
 - Ποιες λειτουργίες / εντολές θα κάνει ο επεξεργαστής
 - Με ποια σειρά θα τις κάνει
- Μη διαδικαστικός Ορισμός αλγορίθμων:
 - Συναρτησιακός προγραμματισμός (Lambda λογισμός, λ-calculus)
 - Λογικός προγραμματισμός (Κατηγορηματική Λογική, Predicate logic)

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 43

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μη-διαδικαστικοί Αργόριθμοι 2/3

- Λίστα
 - [dog cat pig]
 - [3 7 14 2]
 - [elephant]
 - [] κενή λίστα
- head ([a b c d]) = a
- tail ([a b c d]) = [b c d]
- X:L λίστα με head X και tail L
- dog: [cat pig] = [dog cat pig]

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 44

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μη-διαδικαστικοί Αργόριθμοι 2/3

head (X :L)=X
 tail (X :L)=L
 head (L) : tail (L) =L για L ≠ []
 head ([dog]) = dog , tail ([dog]) = []

a:b:c [d]=a:b:[c d] =a:[b c d]=[a b c d]

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 45

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Συναρτησιακός Προγραμματισμός (Functional Programming) 1/2

- Αλγόριθμοι = συνάρτηση
 - Π.χ $y = \text{sum}(L)$ άθροισμα στοιχείων λίστας L
- Βασίζεται στον lambda λογισμό του Church
- Ορισμός Συνάρτησης
 - $\text{Sum}(L) = \text{if } L = [] \text{ then } 0 \text{ else add}(\text{head}(L), \text{sum}(\text{tail}(L)))$
 - $\text{add}(x,y) = x+y$
 - $\text{mult}(L) = \text{if } L = [] \text{ then } 1 \text{ else times}(\text{head}(L), \text{mult}(\text{tail}(L)))$
 - $\text{times}(x,y) = x \cdot y$

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 46

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

(Functional Programming) 2/2

- Γενίκευση Ορισμού – Συναρτήσεις Ανώτερης Τάξης
 - $\text{reduce}(op, base, L) = \text{if } L = [] \text{ then base else } op(\text{head}(L), \text{reduce}(op, base, \text{tail}(L)))$
 - $\text{sum}(L) = \text{reduce}(\text{add}, 0, L)$
 - $\text{mult}(L) = \text{reduce}(\text{time}, 1, L)$
- ή κατά Lambda λογισμό
 - $\text{sum} = \text{reduce}(\text{add}, 0)$
 - $\text{mult} = \text{reduce}(\text{times}, 1)$
 - και $\text{largest} = \text{reduce}(\text{max}, 0)$
 - με $\text{max}(x,y) = \text{if } x \geq y \text{ then } x \text{ else } y$

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 47

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Συναρτησιακό Πρόγραμμα

- Συναρτησιακό Πρόγραμμα = {ορισμός συνάρτησης}
- Γλώσσες συναρτησιακές: Lisp, HOPE, ML, Miranda, κλπ
 - π.χ (Συνάρτηση ταξινόμησης λέξεων κατά αλφαβητική σειρά) ταξινόμηση (L)=if L=[] then [] else εισαγωγή (head(L), ταξινόμηση (tail (L))) (Συνάρτηση εισαγωγής μιας λέξης σε μια ταξινομημένη λίστα L) εισαγωγή (word,L)=if L=[] then [word] else if προηγείται (word, head(L)) then word: L else head(L): εισαγωγή (word, tail (L)) (Συνάρτηση που αποφασίζει εάν word1 >a word2) προηγείται (word1, word2)= if word1 >a word2 then true else false
- Η σειρά των ορισμών δεν έχει σημασία!

Κ. Χαλζάτος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 48

Συναρτησιακό Πρόγραμμα

- Εκτέλεση συναρτησιακού προγράμματος = εφαρμογή αυτού στα δεδομένα εισόδου

application → applicative προγραμματισμός

n.x

ταξινόμηση ([dog pig cat]) =
 εισαγωγή (head ([dog pig cat]),
 ταξινόμηση (tail ([dog pig cat])))
 = εισαγωγή (dog, ταξινόμηση (tail ([dog pig cat])))
 = εισαγωγή (dog, ταξινόμηση ([pig cat]))
 = εισαγωγή (dog, εισαγωγή (head ([pig cat]),
 ταξινόμηση (tail ([pig cat])))
 = εισαγωγή (dog, εισαγωγή (pig, ταξινόμηση (tail ([pig cat])))
 = εισαγωγή (dog, εισαγωγή (pig, ταξινόμηση ([cat])))
 = ...
 = εισαγωγή (dog, [cat pig])
 =
 = [cat dog pig]

Συναρτησιακό Πρόγραμμα

Πρόβλημα : σειρά εφαρμογής των ορισμών

n.x

ταξινόμηση ([dog pig cat]) = ...
 = εισαγωγή (dog, ταξινόμηση ([pig cat]))
 = if προηγείται (dog, head(ταξινόμηση([pig cat]))
 then dog: ταξινόμηση ([pig cat])
 else head(ταξινόμηση([pig cat]):
 εισαγωγή (dog, tail(ταξινόμηση ([pig cat])))

- Αποτέλεσμα: χρειάζεται πιο πολύ δουλειά
- Συμπέρασμα: η σειρά εφαρμογής επιρραζει τον συνολικό χρόνο επεξεργασίας
- Όμως: δεν επηρεάζει το τελικό αποτέλεσμα (υλικό παθολογικών περιπτώσεων) (Το θεώρημα Church – Rosser)
- Πλεονέκτημα: Παραλληλισμός = ταυτόχρονη εφαρμογή πολλών ορισμών

Λογικός Προγραμματισμός – Logic Programming

- Βασίζεται στην **Κατηγορηματική Λογική** (predicate logic)
- Πρόγραμμα = {πρόταση} + στόχος
- Εκτέλεση προγράμματος = **απόδειξη** αληθείας του στόχου
- Γλώσσα : Prolog
- Προτάσεις : head ← body : **κανόνες**
 πουλί (x) ← κάνει - αυγά (x), έχει - φτερά (x).
 ερπετό (x) ← κάνει - αυγά (x), έχει - λέπια (x).
 head ← : **υπενοχές**
 έχει - λέπια (πύθων) ←
 κάνει - αυγά (καναρίνι) ←
 έχει - φτερά (καναρίνι) ←
 ή ← body : στόχος
 ? goal
 ? πουλί (καναρίνι)
 ? ερπετό (πύθων)
 ? πουλί (x)

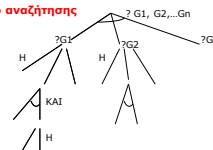
Logic Programming

- Κανόνες συλλογισμού – ανάλυση – resolution

$$\begin{matrix} ?A \\ A \leftarrow B \end{matrix} \} \rightarrow ?B$$

$$\begin{matrix} ?A \\ A \leftarrow \end{matrix} \} \square \text{ true}$$

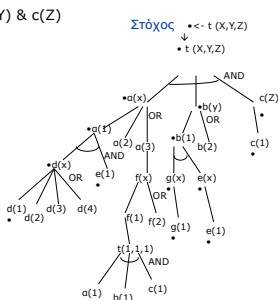
- Ενοποίηση – unification
- Κανόνες αναζήτησης **ΚΑΙ – Η** δένδρο αναζήτησης



- Το δένδρο αναζήτησης διαγράφεται **Κατά Βάθος πρώτα από αισιατά προς τα δεξιά**
- Και γίνεται **οπισθοδρόμηση (backtracking)**, όταν υπάρχει αποτυχία

RESOLUTION TREE AND – OR ΑΝΑΛΥΤΙΚΟ ΔΕΝΤΡΟ

- t (X,Y,Z) ← a(X) & b(Y) & c(Z)
- a(1) ← d(X) & e(1)
- a(2)
- a(3) ← f(x)
- b(1) ← g(x) & e(x).
- b(2).
- c(1).
- d(1).
- d(2).
- d(3).
- d(4).
- e(1).
- f(1) ← t(1,1,1).
- f(2).
- g(1).



RESOLUTION TREE AND – OR ΑΝΑΛΥΤΙΚΟ ΔΕΝΤΡΟ

- ΠΑΡΑΔΕΙΓΜΑΤΑ
- factorial (0,1).
- factorial (N,X) ← (M:=N -1) & factorial (M,Y) & (X:=M*Y).
- factorial (N,X) ← !t(N,0) & write (' N is negative').
- ← factorial (3,X). **ερώτημα**
- ← factorial (3,6). **απάντηση**
- Συμπέρασμα :
 Έλληνας (σωκράτης).
 Έλληνας (δίας).
 άνθρωπος (σωκράτης).
 θνητός (x) ← άνθρωπος (x).
- ← θνητός (x). **ερώτημα**
- ← θνητός (σωκράτης).
- ← ¬ θνητός (x). **ερώτημα**
- ← fa1.
- ← έλληνας (x) & ¬ θνητός (x). **ερώτημα**
- ← έλληνας (δίας) & ¬ θνητός (δίας).

"ΑΡΝΗΣΗ μέσω ΑΠΟΤΥΧΙΑΣ"
 "ΥΠΟΘΕΣΗ ΚΛΕΙΣΤΟΥ ΚΟΣΜΟΥ"

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

RESOLUTION TREE AND - OR ANALYTIKO ΔΕΝΤΡΟ

□ Παραδείγματα

n1) $\text{πουλί}(x) \leftarrow \text{κάνει} - \text{αυγά}(x)$, $\text{έχει} - \text{φτερά}(x)$
 $\text{κάνει} - \text{αυγά}(\text{καναρίνι}) \leftarrow$
 $\text{έχει} - \text{φτερά}(\text{καναρίνι}) \leftarrow$
 $? \text{πουλί}(\text{καναρίνι})$
 $? \text{πουλί}(x)$

απάντηση: **NAI**
 απάντηση $x = \text{καναρίνι}$

n2) $\text{πουλί}(x) \leftarrow \text{κάνει} - \text{αυγά}(x)$, $\text{έχει} - \text{φτερά}(x)$
 $\text{κάνει} - \text{αυγά}(\text{πύθων}) \leftarrow$
 $\text{έχει} - \text{φτερά}(\text{καναρίνι}) \leftarrow$
 $? \text{πουλί}(\text{πύθων})$
 $? \text{πουλί}(\text{καναρίνι})$

απάντηση: **OXI - Fail**
 απάντηση: **OXI - Fail**

n3) $\text{παππούς}(X,Y) \leftarrow \text{γονιός}(X,Z)$, $\text{γονιός}(Z,Y)$
 $\text{γονιός}(\text{Ελένη}, \text{Κώστας}) \leftarrow$
 $\text{γονιός}(\text{Κώστας}, \text{Γιώργος}) \leftarrow$
 $\text{γονιός}(\text{Κώστας}, \text{Χρήστος}) \leftarrow$
 $? \text{παππούς}(\text{Ελένη}, \text{Χρήστος})$

απάντηση: **NAI**

$? \text{παππούς}(\text{Ελένη}, Y)$

1^η απάντηση: **NAI** $Y = \text{Γιώργος}$
 2^η απάντηση: **NAI** $Y = \text{Χρήστος}$

$? \text{παππούς}(X,Y)$

1^η απάντηση: **NAI** $X = \text{Ελένη}$ $Y = \text{Γιώργος}$
 2^η απάντηση: **NAI** $X = \text{Ελένη}$ $Y = \text{Χρήστος}$
 3^η απάντηση: **OXI** **Fail**

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 55

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

RESOLUTION TREE AND - OR ANALYTIKO ΔΕΝΤΡΟ

□ $\leftarrow t$. goal

$t \leftarrow a \ \& \ b \ \& \ c$.
 $a(1) \leftarrow d \ \& \ e$.
 $a(2)$.
 $a(3) \leftarrow f$.
 $b(1) \leftarrow g \ \& \ e$.
 $b(2)$.
 c .
 $d(1)$.
 $d(2)$.
 $d(3)$.
 $d(4)$.
 e .
 $f(1)$.
 $f(2)$.
 g .

Rules & Facts

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 56

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

RESOLUTION TREE AND - OR ANALYTIKO ΔΕΝΤΡΟ

□ Παραδείγμα

n4) $\text{ταξινόμηση}(\text{H:T,S}) \leftarrow \text{ταξινόμηση}(\text{T,L})$, $\text{εισαγωγή}(\text{H,L,S})$
 $\text{ταξινόμηση}(\text{[]}, \text{[]}) \leftarrow$
 $\text{εισαγωγή}(\text{X:H:T}, \text{X:H:T}) \leftarrow \text{προηγείται}(\text{X,H})$
 $\text{εισαγωγή}(\text{X:H:T1}, \text{H:T2}) \leftarrow \text{προηγείται}(\text{H,X})$, $\text{εισαγωγή}(\text{X,T1,T2})$
 $\text{εισαγωγή}(\text{X,[],[X]}) \leftarrow \text{προηγείται}(\text{X,Y}) \leftarrow X > Y$

$? \text{ταξινόμηση}([\text{dog cat pig}], [\text{cat dog pig}])$

απάντηση: **NAI**

$? \text{ταξινόμηση}([\text{dog cat pig}], X)$

απάντηση: $X = [\text{cat dog pig}]$

$? \text{ταξινόμηση}(X, [\text{cat dog pig}])$

απαντήσεις: $[\text{cat dog pig}]$
 $[\text{cat pig dog}]$
 $[\text{cat dog pig dog}]$
 $[\text{pig dog cat}]$

$? \text{ταξινόμηση}(X, [\text{dog cat pig}])$

απάντηση ;

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 57

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

RESOLUTION TREE AND - OR ANALYTIKO ΔΕΝΤΡΟ

□ Η ΔΙΑΤΑΞΗ των προτάσεων «παίζει» ρόλο

□ Η ΔΙΑΤΑΞΗ των goals «παίζει» ρόλο

① predecessor1 (X,Z) : - parent (X,Z).
 predecessor1 (X,Z) : - parent (X,Y), predecessor1 (Y,Z).

② predecessor2 (X,Z) : - parent (X,Y), predecessor2 (Y,Z).
 predecessor2 (X,Z) : - parent (X,Z).

③ predecessor3 (X,Z) : - parent (X,Z).
 predecessor3 (X,Z) : - predecessor3 (X,Y), parent (Y,Z).

④ predecessor4 (X,Z) : - predecessor4 (X,Y), parent (Y,Z).
 predecessor4 (X,Z) : - parent (X,Z).

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 58

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

RESOLUTION TREE AND - OR ANALYTIKO ΔΕΝΤΡΟ

parent (Pam, Bob). % Pam is parent of Bob.
 parent (Tom, Bob).
 parent (Tom, Lig).
 parent (Bob, Ann).
 parent (Bob, Pat).
 parent (Pat, Jim).

?predecessor. (Tom, Pat).

① ② YES
 ④ " More core needed" !

?predecessor. (Lig, Jim).

① ② NO
 ③ ④ " More space needed"!

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 59

Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΓΛΩΣΣΑ PROLOG

□ ΥΠΟΣΥΝΟΛΟ ΤΟΥ ΚΑΤΗΓΟΡΗΜΑΤΙΚΟΥ ΛΟΓΙΣΜΟΥ 1^{ης} ΤΑΞΕΩΣ "ΠΡΟΤΑΣΕΙΣ HORN"

□ ΤΡΕΙΣ ΜΟΡΦΕΣ ΠΡΟΤΑΣΕΩΝ HORN

1. A.	ΓΕΓΟΝΟΤΑ	} ΑΞΙΩΜΑΤΑ =
2. $B \leftarrow A_1 \ \& \ A_2 \ \& \ \dots \ \& \ A_n$	ΚΑΝΟΝΕΣ	
3. $\leftarrow G_1 \ \& \ G_2 \ \& \ \dots \ \& \ G_m$	ΣΤΟΧΟΙ/ ΕΡΩΤΗΜΑΤΑ	} ΘΕΩΡΗΜΑ = ΥΠΟΛΟΓΙΣΜΟΣ

□ ΔΥΟ ΕΡΜΗΝΕΙΕΣ ΤΗΣ $B \leftarrow A_1 \ \& \ A_2 \ \& \ \dots \ \& \ A_n$

1. ΔΗΛΩΤΙΚΗ
 Β ΑΛΗΘΗΣ ΕΑΝ Α₁ ΚΑΙ Α₂ ΚΑΙ...ΚΑΙ Α_n ΑΛΗΘΗ

2. ΔΙΑΔΙΚΑΣΤΙΚΗ
 ΓΙΑ ΝΑ ΑΠΟΔΕΙΞΕΙΣ ΤΗ Β ΑΠΟΔΕΙΞΕ ΠΡΩΤΑ ΤΗΝ Α₁ ΚΑΙ Α₂ ΚΑΙ...Α_n

Κ. Χαλάτσος, Εισαγωγή στην Επιστήμη της Πληροφορικής και των Τηλεπικοινωνιών 60

ΓΛΩΣΣΑ PROLOG

- ΟΙ A, B, A₁...A_n, G₁...G_n ΛΕΓΟΝΤΑΙ ΑΤΟΜΑ
- ΤΑ ΑΤΟΜΑ ΕΙΝΑΙ ΤΗΣ ΜΟΡΦΗΣ

P (t₁, t₂,...,t_n) ΚΑΤΗΓΟΡΗΜΑ η ΟΡΩΝ

t_i ΟΡΟΣ = $\begin{cases} \text{σταθερά} & (15, 'ab', true..) \\ \text{σταθερή συνάρτηση} & f(t'_1, \dots, t'_m) \\ \text{μεταβλητή} & (X, Y, \dots) \end{cases}$

- ΤΟ ΚΑΤΗΓΟΡΗΜΑ ΕΚΦΡΑΖΕΙ ΜΙΑ ΣΧΕΣΗ μεταξύ των ΟΡΩΝ t₁, t₂,...,t_n

ΠΡΟΓΡΑΜΜΑ PROLOG

□ ΠΡΟΓΡΑΜΜΑ PROLOG :

ΕΙΝΑΙ ΜΙΑ ΑΚΟΛΟΥΘΙΑ ΠΡΟΤΑΣΕΩΝ HORN ΜΕ ΚΑΘΟΛΙΚΑ ΠΟΣΟΔΕΙΚΤΟΥΜΕΝΕΣ ΜΕΤΑΒΛΗΤΕΣ

- Η Εκτέλεση του προγράμματος είναι
 - ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΤΟΝ ΣΤΟΧΟ (GOAL DRIVEN)
 - ΑΠΟ ΠΑΝΩ ΠΡΟΣ ΤΑ ΚΑΤΩ (TOP - DOWN)
 - ΠΡΟΣ ΤΑ ΟΠΙΣΘΕΝ ΣΥΛΛΟΓΙΣΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ (BACKWARD REASONING)
- ΜΠΟΡΕΙ ΝΑ ΠΑΡΑΣΤΑΘΕΙ ΜΕ ΕΝΑ AND - OR ΑΝΑΛΥΤΙΚΟ ΔΕΝΔΡΟ
 - ΑΠΟΔΕΙΞΗ
 - ΑΝΑΖΗΤΗΣΗ
- ΠΟΥ ΔΙΑΓΡΑΦΕΤΑΙ ΜΕ ΜΙΑ
 - ΚΑΤΑ ΒΑΘΟΣ ΠΡΩΤΑ- ΑΠΟ ΑΡΙΣΤΕΡΑ ΠΡΟΣ ΤΑ ΔΕΞΙΑ ΣΕΙΡΑ
- ΜΕ
 - ΑΝΑΛΥΣΗ (RESOLUTION) - ΕΝΟΠΙΟΙΣΗ (UNIFICATION) & ΟΠΙΣΤΟΔΡΟΜΗΣΗ (BACKTRACKING)

Η ΓΛΩΣΣΑ PROLOG

- ΑΛΦΑΒΗΤΟ:
 - upper-case letters : A, B,...,Z
 - lower-case letters : a, b,...,g
 - digits : 0,1,...,9
 - special characters : '.,:;...+,-,*,^,/,..._
- ΑΤΟΜΑ :
 - Strings of letters, digits & the underscore '_' με πρώτο lower-case letter
 - Strings of special characters
 - Strings of characters enclosed in single quotes
- ΑΡΙΘΜΟΙ:
 - ακέραιοι
 - reals (όχι πόντα)
- ΜΕΤΑΒΛΗΤΕΣ
 - Strings of letters, digits & underscore '_'
 - Start with upper-case letter or underscore '_' = ανώνυμη μεταβλητή=εμφανίζεται μόνο μια φορά στην πρόταση
 - has child (x) :- parent (X,_)

Η ΓΛΩΣΣΑ PROLOG

□ ΔΟΜΕΣ



n.x date (Day, Month, Year)
point (X,Y)
segment (point (X₁, Y₁), point (X₂,Y₂))

□ ΛΙΣΤΕΣ

- Δομές με functor "[]"
- Παροστάσεις:
 - [a₁, a₂, a₃, a₄]
 - (a₁, -a₂, 'a₃, -(a₄, nil))
 - nil = [] empty list
 - [Head | Tail]
 - [a₁ | Tail]:=[a₁ | [a₂, a₃, a₄]]
 - =-[a₁ | [a₂ | Tail₂]]
 - =-[a₁ | [a₂ | [a₃ | Tail₃]]]
 - =-[a₁,a₂,a₃ | Tail₃]
 - =-[a₁,a₂ | Tail₂]
 - =-[a₁,a₂,a₃,a₄ | nil]

B ← A1 & A2 &...& An
B:- A1, A2,... An

Η ΓΛΩΣΣΑ PROLOG

- ΑΡΙΘΜΗΤΙΚΗ
 - Αριθμητικοί τελεστές + - * / mod , div (υπόλοιπο), (ακέραια διαίρεση)
 - is
 - Συγκρίσεις τιμών (:= διαφορετικό του =)
 - > < >= <= := =\=
 - ? I+A =B+2 → A=2 B=1
 - ? I+A := b+2 No
 - gcd (X,X,X).
 - gcd (X,Y,D) :- X<Y,
Y1 is Y-X,
gcd (X,Y1,D)
 - length ([],0).
 - length ([_ | Tail], N) :- length (Tail,N1),
N is N1+1

Πράξεις επί λιστών

- member (X,L)
- member (X,[X | Tail]).
- member (X,[_ | Tail]) :- member (X, Tail).
- concatenation
 - concat (L1, L2, L3)
 - concat ([], L, L).
 - concat ([X | L1], L2, [X | L3]) :-
concat (L1, L2, L3).
- !! member (X,L):- concat (L1, [X | L2], L).
- !! member (X,L):- concat (L, [X | _], L).
- add (X, L, [X | L])
- del (X, [X | Tail], Tail).
- del (X, [_ | Tail], [_ | Tail1]) :-
del (X, Tail, Tail1).
- !! insert (X, List, Biggerlist) :-
del (X, Biggerlist, List).
- !! member (X,L):- del (X, L, _).

List Permutation

- **1st version**

```
permute (L, [H | T]) :- append (V, [H | U], L),
                        append (V, U, W),
                        append (W, T),
                        permute ([], []).
```
- **2nd version**

```
permute ([], []).
permute (L, [H | T]) :- delete (H, L, L1),
                        permute (L1, T).

append ([], L, L).
append ([H | T], L, [H | L1]) :- append (T, L, L1).

delete (H, [H | T], T).
delete (X, [H | T], [H | U]) :- delete (X, T, U).
```

List Permutation

- **intersection (L1, L2, L).**

```
intersection ([H | T], L, [H | Rem]) :-
    member (H, L), !,
    intersection (T, L, Rem).
intersection ([_ | T], L, Res) :- intersection (T, L, Res).
intersection ([], _, []).
```
- **set-inclusion (L, S).**

```
set-inclusion ([], _).
set-inclusion ([H | T], Set-2) :-
    member (H, Set-2),
    set-inclusion (T, Set-2).
```
- **flatten (List, Flatlist).**

```
flatten ([H | T], Flatlist) :-
    flatten (H, Flat_H),
    flatten (T, Flat_T),
    concat (Flat_H, Flat_T, Flatlist).
flatten ([], []).
flatten (X, [X]).
```

List Permutation

- **sublist**

```
sublist (S,L) :- concat (L1, L2, L),
                 concat (S, L3, L2).
```
- **permutations**
 - i. permutation ([], []).


```
permutation ([X | L], P) :-
    permutation (L, L1),
    insert (X, L1, P).
```
 - ii. permutation ([], []).


```
permutation (l, [X, P]) :- del (X, L, L1),
                             permutation (L1, P).
```

List Permutation

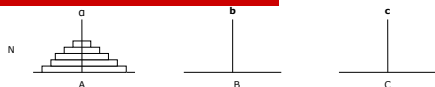
- **reverse**

```
reverse (l, Res) :- rev (l, [], Res).
rev ([Head | Tail], Accu, R) :-
    rev (Tail, [Head | Accu], Res).
rev ([], Res, Res).
```
- **reverse ([], []).** % naive-reverse

```
reverse ([Head | T], L) :-
    reverse (T, R),
    append (R, [Head], L).
```
- **append ([], L, L).**

```
append ([H | T], L, [H | Tail]) :- append (t, L, Tail).
```

TOWERS OF HANOI



Problem: Transfer the disks from peg a to peg b moving one disk at a time (the top disk). No disk should be placed on the top of a smaller disk

- ```
hanoi (1, A, B, C) :- !, write ('Move disk'),
 write (A),
 write ('to'),
 write (B), nl.

hanoi (n, A, b, C) :- m is n-1,
 hanoi (m, A, C, B),
 hanoi (1, A, B, C),
 hanoi (m, C, B, A).

? hanoi (3,a,c,b).
```

## Πρόγραμμα 2 Πίθηκος & μανάνα 1/2



- Initial : state (atdoor, onfloor, atwindow, hasnot)
 

```

graph TD
 state["state (atdoor, onfloor, atwindow, hasnot)"]
 state --> atdoor
 state --> onfloor
 state --> atwindow
 state --> hasnot
 atdoor --- monkey
 onfloor --- monkey
 atwindow --- box
 hasnot --- box

```
- Goal: state ( \_ , \_ , \_ , has).

## Πρόγραμμα 2

### Πιθηκος & μπανάνα 2/2

```

Move : state1 → state2 move (state1, M, state2)
move={grasp_banana, climp_box, push_box, work_around}
move (state (middle, onbox, middle, hasnot),
 grasp,
 state (middle, onbox, middle, has)). % grasp banana
move (state (P, onfloor, P, H),
 climp,
 state (P, onbox, P, H)). % climp box
move (state (P1, onfloor, P1, H),
 push ((P1, P2),
 state (P2, onfloor, P2, H))). % push box
move (state (P1, onfloor, B, H),
 walk (P1, P2),
 state (P2, onfloor, B, H)). % walk from P1 to P2
canget (state (_, _, _, has)).
canget (State1) :- move (State1, Move, state2),
 canget (State2).
? canget (state (atdoor, onfloor, atwindow, hasnot)).

```

## Search tree

