

Ευφυείς Τεχνολογίες – Πράκτορες

**Ενότητα 3:
Αναπαράσταση Γνώσης
και Επίλυση Προβλημάτων**

Δημοσθένης Σταματής
demos@it.teithe.gr
www.it.teithe.gr/~demos

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Μαθησιακοί Στόχοι της ενότητας 3

- ☞ Ποιοι είναι οι τρόποι αναπαράστασης γνώσης που σχετίζονται με ένα πρόβλημα και ποιος ο ρόλος της συλλογιστικής για την παραγωγή νέας γνώσης (βασικές έννοιες - εισαγωγικά)
- ☞ Γενικός ορισμός ενός προβλήματος στα πλαίσια της Τεχνητής Νοημοσύνης (TN)
- ☞ Πως αναπαρίσταται ένα πρόβλημα με τη μορφή του χώρου καταστάσεων
- ☞ Το ρόλο που παίζουν οι αλγόριθμοι αναζήτησης ως μια γενική τεχνική επίλυσης προβλημάτων

2

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

Η ύλη της ενότητας 3

- ❖ Μέθοδοι αναπαράστασης γνώσης
- ❖ Συλλογιστική
- ❖ Περιγραφή Προβλημάτων
- ❖ Ανοιχτός και κλειστός κόσμος προβλήματος
- ❖ Χώρος καταστάσεων - Χώρος αναζήτησης
- ❖ Αλγόριθμοι τυφλής αναζήτησης
- ❖ Αλγόριθμοι ευρετικής αναζήτησης

3

Αναπαράσταση Γνώσης

?

Πως θα μπορούσε ένας πράκτορας να αποθηκεύσει και να χειριστεί τη γνώση που προκύπτει από τις παρακάτω φράσεις:

- ✓ Η Νίκη κρατάει μία ομπρέλα
- ✓ Βρέχει!
- • •
- Η ομπρέλα μπορεί να προστατεύσει κάποιον από τη βροχή
- Η ομπρέλα δεν είναι χρήσιμη όταν φυσάει δυνατά

4

Αναπαράσταση Γνώσης

TN = Αναπαράσταση Γνώσης + Συλλογιστική
(AI = Knowledge Representation + Reasoning)

Συλλογιστική:

Το να παράγουμε πληροφορία που υπονοείται από άλλες πληροφορίες που ήδη υπάρχουν (και έχουν αναπαρασταθεί με κάποια μορφή).

Αναπαράσταση γνώσης:

Οι μορφές αναπαράστασης γνώσης είναι χρήσιμες όταν μπορούμε να δράσουμε επί αυτών συλλογιστικά

5

Αναπαράσταση Γνώσης

Από τα δεδομένα στη γνώση



Σχήμα από (Γεωργούλη, 2015)

6

Είδη Γνώσης (κατά Bar και Feigenbaum)

- ☞ **Σημασιολογική Γνώση (Semantic Knowledge):**
Αναπαράσταση των αντικειμένων ενός κόσμου και οι συσχετίσεις μεταξύ τους
- ☞ **Διαδικαστική Γνώση (Procedural Knowledge):**
Αναπαράσταση/περιγραφή του πώς εκτελείται κάποια εργασία
- ☞ **Επεισοδιακή Γνώση (Episodical Knowledge):**
Η γνώση που βασίζεται σε συμβάντα (events) και στη χρονική σειρά που αυτά συμβαίνουν.
- ☞ **Μετα-γνώση (Meta-knowledge):**
Αναπαράσταση της γνώσης για το πώς θα εφαρμόζονται επιμέρους τμήματα ή διαφορετικές μορφές της γνώσης.

7

Αναπαράσταση Γνώσης: Σημαντικότερες μέθοδοι

Μαθηματική Λογική

- ✓ προτασιακός λογισμός
- ✓ κατηγορηματικός λογισμός
- ✓ **κατηγορηματικός λογισμός πρώτης τάξης**
- ✓ Λογική με χρόνο

Δομημένες Μορφές Αναπαράστασης Γνώσης

- ✓ **σημασιολογικά δίκτυα**
- ✓ εννοιολογικοί χάρτες
- ✓ **πλαίσια**
- ✓ οντολογίες

Συστήματα Κανόνων

8

Λογική

- Μαθηματική Λογική
- Κατηγορηματικός Λογισμός 1^{ης} τάξης
- Λογικός Προγραμματισμός
- Λογική με χρόνο

-
- *parent(kostas, nikos).*
 - *parent(dimitra, nikos).*
 - *parent(kostas, antonis).*
 - *parent(nikos, yanna).*
 - *parent(nikos, vasilis).*
 - *grandparent(X,Y) <= parent(X,Z), parent(Z,Y).*

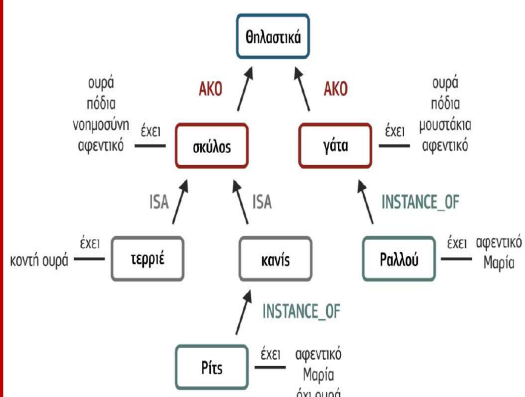
9

Σημασιολογικά Δίκτυα

- Το **σημασιολογικό δίκτυο** είναι μία απλή σχηματική μέθοδος αναπαράστασης γνώσης που βασίζεται σε ένα κατευθυνόμενο γράφο στον οποίο:
 - ❖ Οι κόμβοι αναπαριστούν **αντικείμενα, έννοιες και συμβάντα**
 - ❖ Οι ακμές αναπαριστούν **διμελείς σχέσεις** ανάμεσα στους κόμβους

10

Παράδειγμα ιεραρχικού σημασιολογικού δικτύου



Σχήμα από (Γεωργούλη, 2015) (*)

AKO: A Kind Of

Η σχέση **AKO** υπάρχει μεταξύ κόμβων οντοτήτων που αντιπροσωπεύουν γενικότερες κλάσεις οντοτήτων.

ISA:

Η σχέση **ISA** είναι παρόμοια με τη σχέση **AKO**, με τη διαφορά ότι ο συνδεόμενος κόμβος δεν αντιπροσωπεύει κλάση αλλά υποκλάση οντοτήτων.

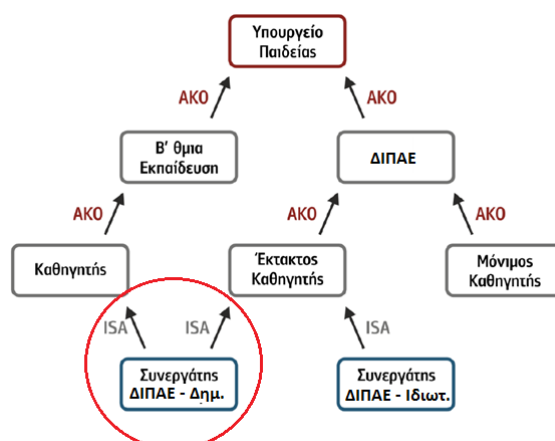
INSTANCE_OF:

Η σχέση **instance_of** υπάρχει μόνο μεταξύ κόμβων οντοτήτων που αποτελούν στιγμιότυπα μίας κλάσης.

(*) <https://repository.kallipos.gr/handle/11419/3381>

13

Παράδειγμα μη ιεραρχικού σημασιολογικού δικτύου



Προσαρμογή σχήματος από (Γεωργούλη, 2015)

14

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Κωδικοποίηση Σημασιολογικού δικτύου σε Prolog

isa(canary, bird).
 isa(ostrich, bird).
 isa(bird, animal).
 isa(opus, penguin).
 isa(robin, bird).
 isa(penguin, bird).
 isa(fish, animal).
 isa(tweety, canary).

hasprop(tweety, color, white).
 hasprop(canary, color, yellow).
 hasprop(bird, travel, fly).
 hasprop(ostrich, travel, walk).
 hasprop(robin, sound, sing).
 hasprop(bird, cover, feathers).

hasproperty(Entity, Property, Value) :-
 hasprop(Entity, Property, Value).

hasproperty(Entity, Property, Value) :-
 isa(Entity, Supentity),
 hasproperty(Supentity, Property, Value).

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ Δ. ΣΤΑΜΑΤΗΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Επεισοδιακή Γνώση: Σημασιολογική αναπαράσταση με χρόνο

Δομή Κίρκε ως αυτόματο πεπερασμένων καταστάσεων για το όχημα του αεροδρομίου.

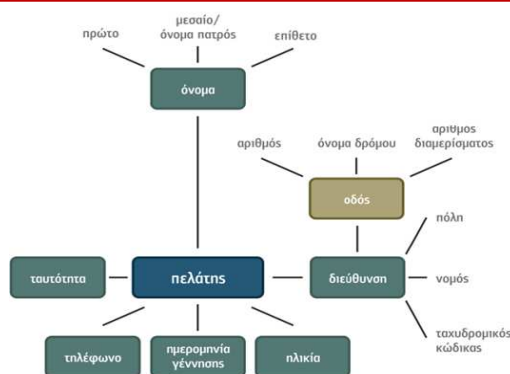
Σχήμα από (Βλαχάβας κ.α., 2020)

16

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ Δ. ΣΤΑΜΑΤΗΣ

Οντολογίες (Ontologies)

Μια **οντολογία** είναι μια τυπική αναπαράσταση ενός συνόλου εννοιών εντός **ενός τομέα** και των σχέσεων μεταξύ των εννοιών αυτών.

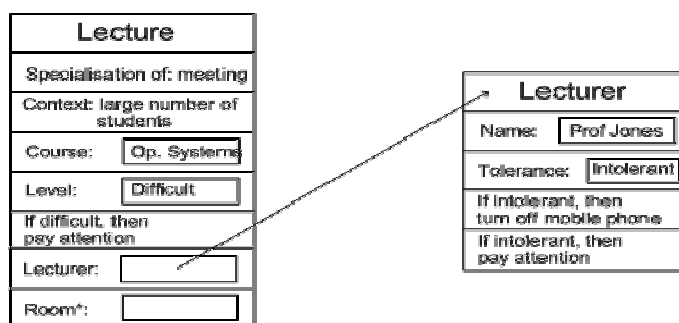


Σχήμα από (Γεωργούλη, 2015)

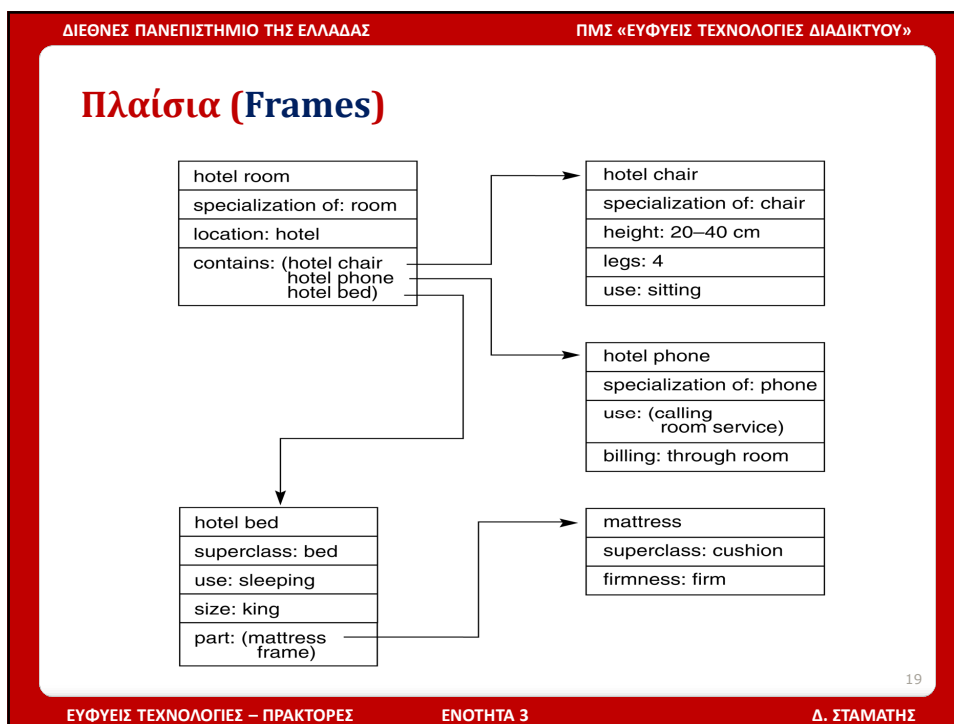
17

Πλαίσια (Frames)

- Ένα πλαίσιο αποτελεί μία **δυναμική** αναπαράσταση μίας έννοιας.
- Αποτελείται από **πεδία (slots)** τα οποία δηλώνουν ιδιότητες ανάμεσα σε αντικείμενα.
- Μερικά πεδία μπορεί να δημιουργούνται **αυτόματα** και άλλα μπορεί αρχικά να είναι **κενά** για να πάρουν τιμή όταν αυτή γίνει γνωστή.
- Μερικά πεδία μπορεί να είναι **σύνδεσμοι** σε άλλα πλαίσια.



18



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Αναπαράσταση Γνώσης με Κανόνες

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

Πίνακας από (Γεωργούλη, 2015)

20

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

Συστήματα κανόνων

- ⇒ Κανόνας 1
 if <η μηχανή τροφοδοτείται με βενζίνη> and
 <η μηχανή γυρίζει>
 then <πρόβλημα με τα μπουζι>
- ⇒ Κανόνας 2
 if <η μηχανή δεν γυρίζει> and
 <τα φώτα δεν ανάβουν>
 then <πρόβλημα με τη μπαταρία> or
 <πρόβλημα με τα καλώδια>
- ⇒ Κανόνας 3
 if <η μηχανή δεν γυρίζει> and
 <τα φώτα ανάβουν>
 then <πρόβλημα με τη μίζα>
- ⇒ Κανόνας 4
 if <υπάρχει βενζίνη στο ντεπόζιτο> and
 <υπάρχει βενζίνη στο καρμπυρατέρ>
 then <η μηχανή τροφοδοτείται με βενζίνη>

21

Κανόνες και Είδη Συλλογιστικής

Συνεπαγωγική συλλογιστική (deductive reasoning): Εξάγει συμπεράσματα βασισμένη 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.

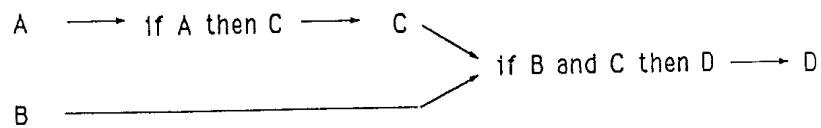
Δεδομένου του κανόνα: Όλα τα σκυλιά του Κώστα είναι καφέ
 και του αποτελέσματος: Τα σκυλιά είναι καφέ
 Υπόθεση που γίνεται: Αυτά τα σκυλιά είναι του Κώστα

(Πίνακας από Βλαχάβας κ.α., 2020)

22

Συστήματα κανόνων - Ορθή συλλογιστική

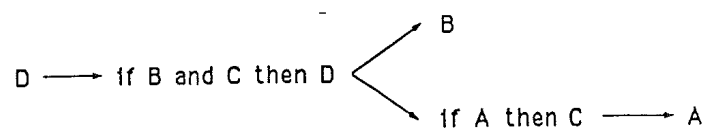
A (is true)
 B (is true)
if A then C
if B and C then D



23

Συστήματα κανόνων - Ανάστροφη συλλογιστική

A (is true)
 B (is true)
if A then C
if B and C then D



24

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Αναπαράσταση Γνώσης

Από τα

- τη λογική
- τα σημασιολογικά δίκτυα,
- τις οντολογίες, και
- τους κανόνες στο

Σημασιολογικό Ιστό

Trust

Proof

Logic

Rules / Query

Ontology

RDF Model & Syntax

XML Query

XML Schema

XML

Namespaces

URI / IRI

Unicode

Signature

Encryption

25

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Γενική Περιγραφή Προβλήματος

➤ Για να γίνει δυνατή η επίλυση ενός **προβλήματος** στην Τεχνητή Νοημοσύνη (TN) απαιτείται ένας τυποποιημένος και σαφής ορισμός

➤ Θεωρούμε ότι ένα **πρόβλημα** μπορεί να οριστεί αν:

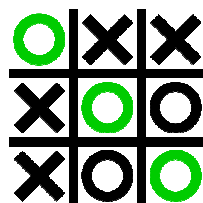
- ✓ Υπάρχει μια δεδομένη **αρχική κατάσταση**
- ✓ Υπάρχει μια επιθυμητή **τελική κατάσταση**
- ✓ Είναι γνωστές κάποιες **ενέργειες** που πρέπει να γίνουν για να προκύψει η επιθυμητή κατάσταση

26

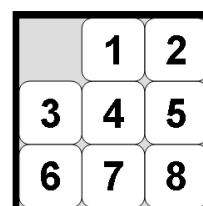
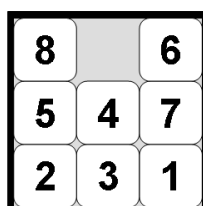
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

Ενδεικτικά Προβλήματα

Τριάρα
(tic-tac-toe)



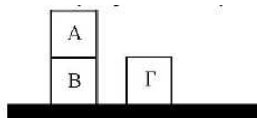
8-puzzle
(N-puzzle)



27

Ενδεικτικά Προβλήματα

Ο κόσμος των κύβων



Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος Α	Κύβος Α είναι ελεύθερος	Κύβος Α πάνω στον κύβο Β
Κύβος Β	Κύβος Γ είναι ελεύθερος	Κύβος Β πάνω στο Τ
Κύβος Γ	Τ έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο Τ
Τ είναι Τραπέζι	Κύβος Β δεν είναι ελεύθερος	

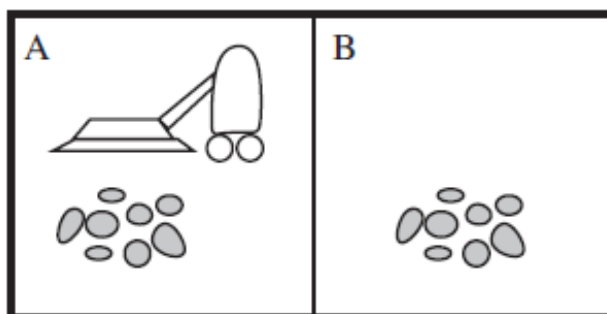
- ❖ Κόσμος του προβλήματος: Τρεις κύβοι και ένα τραπέζι.
- ❖ Μια Κατάσταση:

Κύβος Α πάνω στον κύβο Β
Κύβος Β πάνω στο Τ
Κύβος Γ πάνω στο Τ
Κύβος Α ελεύθερος
Κύβος Γ ελεύθερος

28

Ενδεικτικά Προβλήματα

Η σκούπα Robot που καθαρίζει τα δύο δωμάτια A και B



29

Ενδεικτικά Προβλήματα

Ιεραπόστολοι και Κανίβαλοι

Στην όχθη ενός ποταμού βρίσκονται **3 ιεραπόστολοι** και **3 κανίβαλοι**, οι οποίοι επιθυμούν να το διασχίσουν για να βρεθούν στην άλλη όχθη

- Στη διάθεσή τους έχουν 1 βάρκα που χωράει το πολύ 2 άτομα
- Αν σε κάποια όχθη βρεθούν περισσότεροι κανίβαλοι τρώνε τους ιεραπόστολους !



30

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Το Πρόβλημα του ταξιδεύοντος εμπόρου

Οδικός Χάρτης Β. Ελλάδας

31

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ
ΕΝΟΤΗΤΑ 3
Δ. ΣΤΑΜΑΤΗΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Το Πρόβλημα του ταξιδεύοντος εμπόρου

City 1	City 2	Distance (km)
Φλώρινα	Καστοριά	71
Φλώρινα	Έδεσσα	77
Καστοριά	Έδεσσα	91
Έδεσσα	Κιλκίς	50
Έδεσσα	Βέροια	44
Έδεσσα	Θεσσαλονίκη	72
Κιλκίς	Σέρρες	84
Κιλκίς	Θεσσαλονίκη	62
Σέρρες	Δράμα	69
Σέρρες	Καβάλα	154
Δράμα	Ξάνθη	89
Ξάνθη	Κομοτηνή	50
Κομοτηνή	Αλεξανδρούπολη	55
Βέροια	Κοζάνη	61
Βέροια	Κατερίνη	70
Κοζάνη	Κατερίνη	121
Κοζάνη	Γρεβενά	47
Κατερίνη	Λάρισα	87
Γρεβενά	Ιωάννινα	100
Ιωάννινα	Καρδίτσα	56
Καρδίτσα	Λάρισα	56

Ένας έμπορος πρέπει να επισκεφτεί όλες τις πόλεις της Β. Ελλάδας κάνοντας τα λιγότερα δυνατά χιλιόμετρα

Γράφημα Πόλεων Αποστάσεις από το www.apostaseis.gr

32

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ
ΕΝΟΤΗΤΑ 3
Δ. ΣΤΑΜΑΤΗΣ

Κόσμος ενός προβλήματος

- Ο **Κόσμος** ενός προβλήματος (***problem world***) ορίζεται από τα αντικείμενα (ή τις οντότητες) που τον αποτελούν, τις ιδιότητες των αντικειμένων και τις σχέσεις που τα συνδέουν.
- Ο **Κόσμος** ενός προβλήματος χαρακτηρίζεται **ως κλειστός** (***closed world***) όταν κανένα νέο αντικείμενο, ιδιότητα ή σχέση δεν μπορεί να προστεθεί ή να αφαιρεθεί. [είναι στατικός]
- Σε αντίθετη περίπτωση χαρακτηρίζεται **ως ανοιχτός** (***open world***).
[Η περιγραφή του προβλήματος μπορεί να αλλάζει δυναμικά]

33

Ορισμός Προβλήματος με Χώρο Καταστάσεων (1)

- **Κατάσταση** ενός κόσμου προβλήματος είναι ένα **στιγμιότυπο** (***instance***) που παράγεται σε μία χρονική στιγμή κατά την εξέλιξη του κόσμου.
- Οι καταστάσεις ενός κόσμου **συνδέονται μεταξύ τους** με την έννοια ότι από μία κατάσταση, κατά την επόμενη χρονική στιγμή, μπορούν να προκύψουν μία ή περισσότερες καταστάσεις
- Θεωρούμε ότι μία κατάσταση προκύπτει από μία άλλη με την εφαρμογή ενός **τελεστή μετάβασης** (***transition operator***)

34

Ορισμός Προβλήματος με Χώρο Καταστάσεων (2)

Ένα πρόβλημα **P** ορίζεται ως μία τετράδα $P = \{ S, I, T, G \}$, όπου:

- **S** είναι ο **χώρος καταστάσεων** (το σύνολο όλων των καταστάσεων)
- **I** είναι μία **αρχική κατάσταση** (ανήκει στο **S**)
- **T** είναι το σύνολο των **τελεστών μετάβασης**
- **G** είναι το σύνολο των **τελικών καταστάσεων** (υποσύνολο του **S**)

Λύση ενός προβλήματος $P = \{ S, I, T, G \}$ είναι μία ακολουθία από τελεστές μετάβασης

$\langle t_1, t_2, t_3, \dots, t_n \rangle$

ώστε να ισχύει:

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

όπου **g** μία τελική κατάσταση του συνόλου **G**

35

Χώρος Αναζήτησης ενός Προβλήματος

- Δοθέντος ενός προβλήματος $P = \{ S, I, T, G \}$,

Χώρος Αναζήτησης (Search Space) SP είναι το σύνολο όλων των καταστάσεων που είναι **προσβάσιμες** από την αρχική κατάσταση **I**.

- Ο χώρος αναζήτησης είναι **υποσύνολο** του χώρου καταστάσεων, καθώς το σύνολο **SP** εξαρτάται από την αρχική κατάσταση **I** ενώ το **S** όχι.

36

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Χώρος Καταστάσεων (1)

Ο Χώρος Καταστάσεων του προβλήματος «ιεραπόστολοι & κανίβαλοι»

37

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Χώρος Καταστάσεων (2)

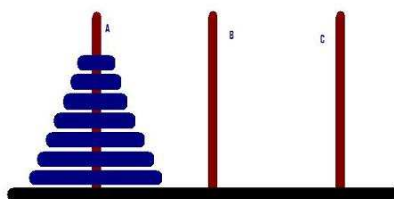
Ο Χώρος Καταστάσεων του προβλήματος «της σκούπας Robot»

38

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

Περιγραφή προβλήματος με αναγωγή

- Μία ακολουθία από τελεστές **ανάγουν** συνεχώς τα προβλήματα σε άλλα «απλούστερα» έως ότου τα υπο-προβλήματα που προκύπτουν τελικά να είναι **άμεσα επιλύσιμα**.
- Παράδειγμα οι **Πύργοι του Hanoi**: Ένας αριθμός δίσκων σε φθίνουσα διάταξη βρίσκονται σε ένα στύλο και πρέπει να μεταφερθούν σε έναν άλλο με την ίδια διάταξη με τους εξής περιορισμούς:
 - ✓ Επιτρέπεται να μετακινείται ένας δίσκος τη φορά
 - ✓ Δεν επιτρέπεται η τοποθέτηση μεγαλύτερου δίσκου σε μικρότερο
 - ✓ Ένα τρίτος στύλος μπορεί να χρησιμοποιηθεί ως βοηθητικός



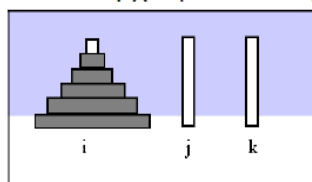
39

Περιγραφή προβλήματος με αναγωγή

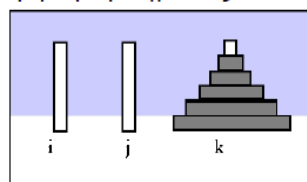
Η λύση του προβλήματος «Πύργοι του Hanoi»

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 Περιγραφή



Τελική Περιγραφή

40

Αλγόριθμοι Αναζήτησης

- Υπάρχουν πολλά προβλήματα της ΤΝ για τα οποία δεν μπορεί να μας δοθεί ένας ειδικός αλγόριθμος για τη λύση τους, αλλά μόνο μία περιγραφή της λύσης. Στην περίπτωση αυτή πρέπει να ψάξουμε να βρούμε τη λύση!
- Πολλά προβλήματα της ΤΝ μπορούν να αναπαρασταθούν αφαιρετικά με τη βοήθεια ενός γραφήματος. Στην περίπτωση αυτή η επίλυση του προβλήματος ταυτίζεται με την αναζήτηση ενός μονοπατιού του γραφήματος
- Στις περισσότερες περιπτώσεις ξεκινούμε από έναν αρχικό κόμβο του γραφήματος και αναζητούμε το μονοπάτι που θα μας οδηγήσει σε έναν κόμβο στόχο

41

Αλγόριθμοι Αναζήτησης

Η γενική ιδέα (1):

- Δοθέντος ενός γράφου, ενός αρχικού κόμβου και ενός κόμβου στόχου :
 - ❑ Εξερευνούμε επαυξανόμενα μονοπάτια από τον αρχικό κόμβο
 - ❑ Διατηρούμε ένα σύνολο-μέτωπο (frontier set) από κόμβους που πρόκειται να εξερευνησουμε άμεσα.
 - ❑ Καθώς η αναζήτηση προχωρά το σύνολο-μέτωπο επεκτείνεται μέχρις ότου βρεθούμε στον κόμβο στόχο.
- Ο τρόπος που επεκτείνουμε και επεξεργαζόμαστε το σύνολο-μέτωπο καθορίζει τη στρατηγική της αναζήτησης

42

Δέντρο Αναζήτησης (OR Δέντρο)

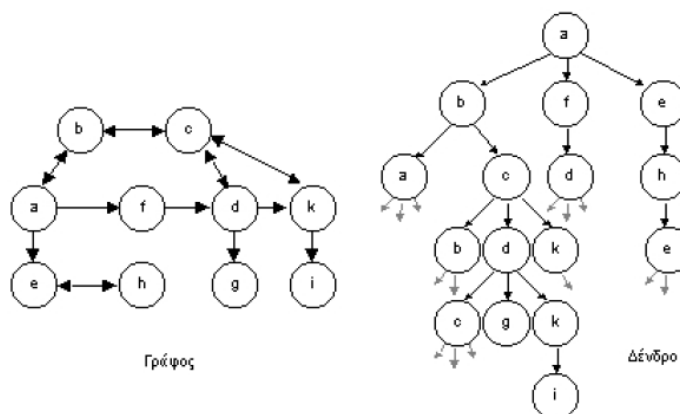
- Για τις ανάγκες των αλγορίθμων ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με τη βοήθεια ενός **OR Δέντρου**

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).

43

Δέντρο Αναζήτησης (OR Δέντρο)

- Η μετατροπή ενός γράφου σε δέντρο αναζήτησης είναι πάντα εφικτή, εμπεριέχει όμως τον κίνδυνο το δέντρο να αποκτήσει κλαδιά με άπειρο μήκος (**συνδυαστική έκρηξη!**)



44

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

OR Δέντρο της Σκούπας Robot

The diagram shows an OR tree for a vacuum robot. The root node is 1. From node 1, two actions are possible: Suck and Right. Sucking leads to node 7 (GOAL) or node 5. From node 5, Suck leads to node 1 (LOOP) and Right leads to node 6. From node 6, Suck leads to node 8 (GOAL) and Left leads to node 1 (LOOP). From node 1, the Right action leads to node 2. From node 2, Left leads to node 1 (LOOP) and Suck leads to node 8 (GOAL) or node 4. Node 4 has two outgoing arrows, indicating further actions.

45

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΑΣ ΠΜΣ «ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ»

Αλγόριθμοι Αναζήτησης

Η γενική ιδέα (2):

The diagram illustrates a search tree. The root node is green and labeled 'αρχικός κόμβος'. Black nodes represent 'κόμβοι που έχουμε επισκεφτεί'. Red nodes represent the 'σύνολο μέτωπο' (frontier). Blue nodes represent 'κόμβοι που δεν έχουμε επισκεφτεί'. A yellow node is labeled 'κόμβος στόχος' (goal node).

46

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ – ΠΡΑΚΤΟΡΕΣ ΕΝΟΤΗΤΑ 3 Δ. ΣΤΑΜΑΤΗΣ

Γενικός Αλγόριθμος Αναζήτησης

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

47

Γενικός Αλγόριθμος Αναζήτησης

```

algorithm general(InitialState, FinalState)
begin
  Closed ← ∅;
  Frontier ← <InitialState>;
  CurrentState ← First(Frontier);
  while CurrentState ≠ FinalState do
    Frontier ← delete(CurrentState, Frontier);
    if CurrentState ∉ ClosedSet then
      begin
        Next ← Expand(CurrentState);
        Frontier ← insert(Next, Frontier);
        Frontier ← prune(Frontier);
        Frontier ← reorder(Frontier);
        Closed ← Closed ∪ {CurrentState};
      end;
    if Frontier = ∅ then return failure;
    CurrentState ← First(Frontier);
  endwhile;
end.

```

48

Αλγόριθμοι Τυφλής Αναζήτησης

- Οι αλγόριθμοι τυφλής αναζήτησης εφαρμόζονται όταν **δεν έχουμε επαρκείς πληροφορίες** που να μας επιτρέπουν την «έξυπνη» εκτίμηση των επόμενων καταστάσεων ενός προβλήματος.

Όνομα Αλγορίθμου	Συνομογραφία	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
Beam Search	BS	Ακτινωτή Αναζήτηση

49

Αλγόριθμοι Ευρετικής Αναζήτησης

- Μία ευρετική τεχνική βασίζεται σε κριτήρια με βάση τα οποία μπορούμε να επιλέξουμε ανάμεσα σε πολλές επιλογές την πιο αποτελεσματική, η οποία θα μας οδηγήσει στο στόχο μας. Συνήθως η ευρετική τεχνική υλοποιείται με μία ευρετική συνάρτηση.
- Μία ευρετική τεχνική:
 - ❑ Βελτιώνει την αποτελεσματικότητα της διαδικασίας αναζήτησης, ενδεχομένως θυσιάζοντας την πληρότητα των λύσεων
 - ❑ Δεν διασφαλίζει ότι θα βρούμε τη βέλτιστη λύση αλλά σχεδόν πάντα βρίσκει μία ικανοποιητική λύση

50

Αλγόριθμοι Ευρετικής Αναζήτησης

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

Αλγόριθμοι Αναζήτησης με αντίπαλο (π.χ. παιχνίδια 2 παικτών)

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

51

Web Links

State problems

<http://www.plastelina.net/game2.html>

<http://www.plastelina.net/game1.html>

Breadth-First Search (BFS)

Erik Demaine, MIT Open Courses

<http://www.youtube.com/watch?v=s-CYnVz-uh4>

<http://www.youtube.com/watch?v=we2xFCpkH0Y>

Depth-First Search (DFS), Topological Sort

Erik Demaine, MIT Open Courses

<http://www.youtube.com/watch?v=AfSk24UTFS8>

How Google makes improvements to its search algorithm

<http://www.youtube.com/watch?v=J5RZOU6vK4Q>

Shortest Path using Dijkstra's Algorithm

<http://www.youtube.com/watch?v=WN3Rb9wVYDY>

Computational Complexity

http://www.youtube.com/watch?v=moPtWq_cVH8

52