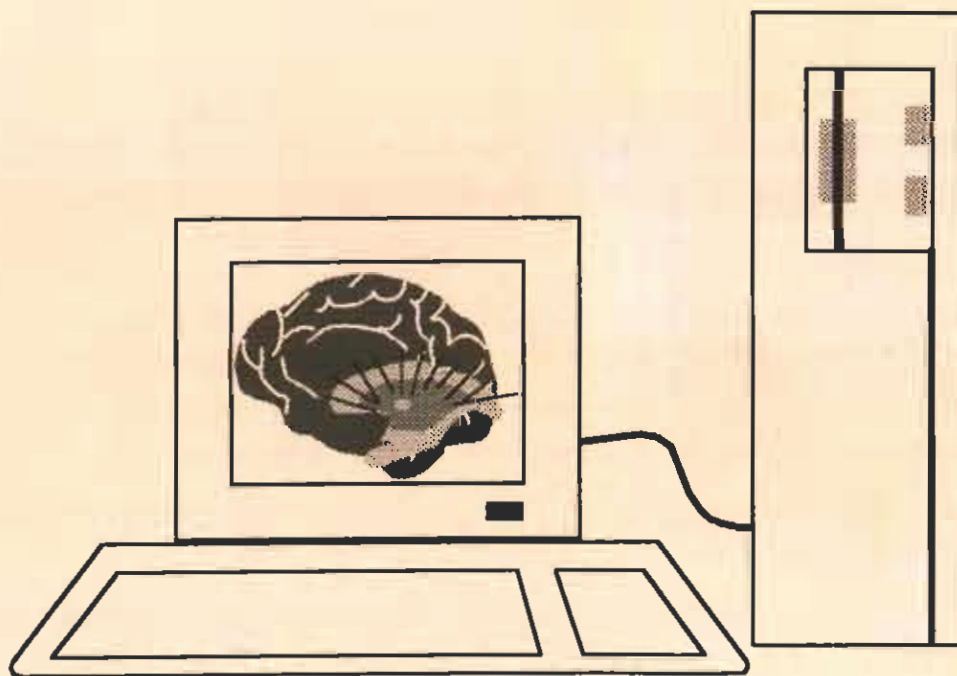


Τ.Ε.Ι. ΠΑΤΡΑΣ  
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ  
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΣΚΕΠΤΟΜΕΝΟΙ ΥΠΟΛΟΓΙΣΤΕΣ :

**ΜΥΘΟΣ Ή ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ ;**



ΚΑΘΗΓΗΤΡΙΑ: ΑΝΤΩΝΟΠΟΥΛΟΥ ΗΡΑ

ΣΠΟΥΔΑΣΤΡΙΕΣ:

ΒΟΥΛΓΑΡΗ ΑΠΟΣΤΟΛΙΑ  
ΠΡΑΤΖΟΥ ΓΙΑΝΝΟΥΛΑ  
ΣΤΡΙΚΟΥ ΜΑΡΙΑ



ΠΑΤΡΑ 1996

ΑΡΙΘΜΟΣ  
ΕΙΣΑΓΩΓΗΣ

1771



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΜΕΡΟΣ Α	1
ΕΙΣΑΓΩΓΗ	2
ΙΣΤΟΡΙΚΟ ΤΗΣ ΤΝ	4
ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ κ ' PROLOG	5
— Α. Η λογική και προγραμματισμός	12
— Β. Ο λογικός προγραμματισμός	16
— Γ. ιστορική εξέλιξη	18
ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	21
1. στοιχεία του προγράμματος	21
2. Τα γεγονότα	23
3. Οι κανόνες	24
4. Τα ερωτήματα	26
ΣΥΝΤΑΞΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ	29
— Τα αντικείμενα της Prolog	29
— Το αλφάβητο της Prolog	29
— Οι μεταβλητές	30
— Οι σταθερές	31
— Τα κατηγορήματα	32
ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ ΚΑΙ Η ΔΙΑΔΙΚΑΣΙΑ — ΕΠΑΝΑΔΡΟΜΗΣΗΣ	35
ΣΥΣΤΗΜΑΤΑ - ΕΜΠΕΙΡΟΓΝΩΜΟΝΕΣ	41
— Συστήματα - Εμπειρογνώμονες και διαχείριση γνώσεων	41
— Ένα Σύστημα - Εμπειρογνώμονας για νεφρικές παθήσεις	44

ΜΕΡΟΣ Β	51
ΕΦΑΡΜΟΓΗ ΕΞΥΠΝΩΝ ΥΠΟΛΟΓΙΣΤΩΝ	52
ΚΑΤΑΛΟΓΟΙ ΕΦΑΡΜΟΓΩΝ	53
1. ΓΕΩΡΓΙΑ	53
2. ΧΗΜΕΙΑ	54
3. COMPUTER SYSTEMS	60
4. ΗΛΕΚΤΡΟΝΙΚΑ	65
5. ΜΗΧΑΝΟΛΟΓΙΑ	75
6. ΓΕΩΛΟΓΙΑ	79
7. INFORMATION MANAGEMENT	82
8. LAW (ΝΟΜΙΚΗ)	86
9. MANUFACTURING (ΚΑΤΑΣΚΕΥΗ)	91
10. ΜΑΘΗΜΑΤΙΚΑ	93
11. MEDICINE (ΙΑΤΡΙΚΗ)	95
12. METEOROLOGY (ΜΕΤΕΩΡΟΛΟΓΙΑ)	121
13. MILITARY SCIENCE	121
14. ΦΥΣΙΚΗ	125
15. PROCESS CONTROL	125
16. SPACE TECHNOLOGY	126

## **ΜΕΡΟΣ Α**

## ΕΙΣΑΓΩΓΗ.

Τα τελευταία 50 χρόνια οι επιστήμονες - ερευνητές που ασχολούνται με τα θέματα της πληροφορικής έθεσαν στο “χώρο τους” το εξής ερώτημα. Μπορούμε να κάνουμε το μηχάνημα “Νοήμονα” ή αλλιώς μπορούμε να κατασκευάσουμε Τεχνητή Νοημοσύνη (Artificial intelligence)?

**Ερώτηση:** *Τι είναι Τεχνητή Νοημοσύνη (Artificial Intelligence).*

Πριν 25 χρόνια δινόταν η εξής απάντηση:

Η Τ.Ν. ασχολείται με την κατασκευή μιας Μηχανής (Ρομπότ) καθ’ ομοίωση του ανθρώπου, η οποία θα έχει την δική της παιδική ηλικία, θα μπορεί να μαθαίνει γλώσσες όπως ένα παιδί και θα επαυξάνει τις γνώσεις της από την επικοινωνία της με το περιβάλλον.

**Παρατήρηση:** Οι παραπάνω θέσεις σήμερα δεν ισχύουν διότι υπήρξαν τα εξής προβλήματα:

- 1) Η νοημοσύνη έχει μία σχετική σημασία και δεν είναι μετρήσιμη..
- 2) Υπάρχουν πράγματα που τα μαθαίνει ο άνθρωπος ακριβώς γιατί διαθέτει έναν ανθρώπινο οργανισμό
- 3) Υπάρχουν πράγματα που εμπίπτουν στην αντίληψη του ανθρώπου ακριβώς από το γεγονός ότι οι άλλοι άνθρωποι τον βλέπουν σαν άνθρωπο.
- 4) Ακόμα και οι φυσιολόγοι και άλλοι ειδικοί δεν μπορούν να εξηγήσουν με σαφήνεια όλες τις συνδέσεις των νευρικών κυττάρων του ανθρώπινου εγκεφάλου και έτσι δυσκολεύουν τα πράγματα για τους μηχανικούς που θέλουν να “μμηθούν” τους ανθρώπινους νευρώνες (νευρικά δίκτυα)

Σήμερα δίνεται μια πιο προσγειωμένη θέση που ταιριάζει καλύτερα στους φυσικούς περιορισμούς του ανθρώπου:

Σκεπτόμενοι Υπολογιστικά, Μύθος ή Πραγματικότητα;

-Η T.N. είναι η μελέτη των πνευματικών ικανοτήτων δια μέσου της χρήσης υπολογιστικών μοντέλων.

ή

-Η T.N ασχολείται με την μηχανιστική (Computer) παράσταση και επεξεργασία των ανθρώπινων δομών γνώσης, τις οποίες προσπαθεί να χρησιμοποιήσει σαν βασικά στοιχεία (μοντέλο) πολύπλοκων συστημάτων (διαδικασιών ) επεξεργασίας πληροφοριών.

**Υπόθεση:(v.Neuman 1955).**

“Ο τρόπος με τον οποίο εργάζεται ο εγκέφαλος είναι όμοιος με τον τρόπο που εργάζεται ο Computer”.

**ΚΕΝΤΡΙΚΟ ΔΟΓΜΑ:(fundamental working assumption).**

“Ο,τι κάνει ο εγκέφαλος πρέπει να θεωρηθεί ως προ κάποιο επίπεδο σαν ένα είδος υπολογισμού”.

Αυτό το είδος υπολογισμού είναι αντικείμενο της παράδοσης.

**Παρατήρηση:** Οι T.N. - ειδικοί διαχωρίζουν σχεδόν πάντοτε την T.N. από την επιστήμη των Computers (computer science). Αν και η T.N. διερευνά τις ιδέες που μπορούν να κάνουν τον computer εξυπνότερο (δηλ. χρησιμότερο).

Οι T.N.- ειδικοί καταφέρανε να συστηματοποιήσουνε κάπως την χρήση των Computers σε πιο γοητευτικές και εντυπωσιακές εφαρμογές, όπως είναι: συστήματα εμπειρογνομώνων (expert systems, θα αναφερθούμε σ' αυτές τις “έξυπνες εφαρμογές” αναλυτικά), κατανόηση εικόνας, Ρομποτική, κατανόηση φυσικής γλώσσας, σχεδιασμός αγοράς, έλεγχος παραγωγής, αυτόματη μετάφραση

## ΙΣΤΟΡΙΚΟ ΤΗΣ Τ.Ν.

α) Ευρωπαϊκή λογοτεχνία. Ο Δρ. Φρανκενστάιν κατασκευάζει έναν τεχνητό άνθρωπο.

β) **George Boole** (1815-1864) (ο πατέρας της άλγεβρας του Boole).

“Οι νόμοι που πρέπει να μελετήσουμε είναι νόμοι από τις πιο σπουδαίες διανοητικές ικανότητές μας. Τα μαθηματικά που πρέπει να αναπτύξουμε είναι μαθηματικά για την ανθρώπινη νοημοσύνη”

Στο βιβλίο του *Au Inveshgahom of the Laws of Thought on which Are founded the Mathematical Theories of Logic and Probabilities* (1854).

γ) **Alan Turing** (1912-1954)

Στο άρθρο του “Computing Machinery and Intelligence” υποστηρίζει ότι “το computing μπορεί να προγραμματισθεί έτσι ώστε να επιδειξει έξυπνη συμπεριφορά”

δ) Η **Darmouth Conference** (1956). Οργανώθηκε από τον John McCarthy.

Είναι η πρώτη φορά που χρησιμοποιήθηκε ο όρος “Τεχνητή νοημοσύνη”. Στην δήμενη αυτή επιστημονική συνάντηση έλαβαν μέρος πρωτεργάτες της Τ.Ν. Mc.Carthy, M.Minshy, Allen Newell, H.Simou κ.α. και εργάστηκαν πάνω στην υπόθεση: “κάθε εκδοχή της μάθησης και κάθε χαρακτηριστικό της νοημοσύνης μπορεί κατά βάση να περιγραφεί ακριβώς ώστε να δύναται να εξομοιωθεί από μία μηχανή”.



## ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ & PROLOG

Ο βασικός μας στόχος (των ερευνητών της περιοχής) είναι να μπορούν με κάποια μέθοδο να δίνουν στον υπολογιστή (μηχάνημα) να “καταλαβαίνει” την ανθρώπινη γλώσσα. Στην συνέχεια θα πρέπει να τον εφοδιάσουμε με την κατάλληλη μέθοδο με την οποία θα χρησιμοποιεί σωστά την γνώση που του δώσαμε (προτάσεις ανθρώπινης γλώσσας) και εξάγει λογικά συμπεράσματα.

Για τον σκοπό αυτό σ’ αυτό το κεφάλαιο θα σας παρουσιάσουμε τον Λογικό Προγραμματισμό και πιο συγκεκριμένα την γλώσσα PROLOG. (Programming in Logic).

Σκοπός μας τώρα είναι να παρουσιάσουμε το Λογικό προγραμματισμό(άμεση εφαρμογή στοιχείων της Μαθηματικής).

Ο όρος “Γλώσσα Προγραμματισμού” βέβαια δεν αποδίδει ακριβώς την πραγματικότητα. Η Prolog, όπως η Formant ή η Pascal, δεν είναι γλώσσες, είναι μόνο ένα τυπικό, ένας φορμαλισμός (notation, formalism) [SD90], με τον οποίο συμβολίζουμε και τυποποιούμε δεδομένα και διαδικασίες. Οι ονομαζόμενες “Γλώσσες Προγραμματισμού” απέχουν πολύ από τις φυσικές γλώσσες και ως προς τις εκφραστικές δυνατότητες και ως προς τη χρήση τους και ως προς την ερμηνεία τους. Θα συνεχίσουμε να χρησιμοποιούμε τον όρο “Γλώσσα Προγραμματισμού”, επειδή είναι καθιερωμένος όρος, εννοώντας πάντοτε το σύνολο των εκφράσεων που εμφανίζονται σε κάθε είδος προγραμματισμού.

Ο κύριος στόχος της ΛΠ και της ΛΚ είναι ο καθορισμός μαθηματικών μοντέλων που επιτρέπουν την απόδειξη της αλήθειας ή του ψεύδους μιας πρότασης από ένα σύνολο άλλων προτάσεων ή υποθέσεων. Μέσα σ' αυτό το πλαίσιο έχουμε ήδη ασχοληθεί με τα μοντέλα που περιγράφουν τον κανόνα της απόθεσης, τις αποδείξεις με πίνακες και την αποδεικτική μέθοδο της επίλυσης.

Η τελευταία αυτή μέθοδος αποτελεί τον πιο ισχυρό δεσμό μεταξύ της Μαθηματικής Λογικής και του Λογικού Προγραμματισμού και παράλληλα γεφυρώνει, κατά κάποιον τρόπο, τα αφηρημένα μαθηματικά μοντέλα με την λειτουργία μιας γλώσσας προγραμματισμού, όπως είναι η Prolog.

Στο πρώτο κεφάλαιο μελετήσαμε τη δυαδική επίλυση σαν αποδεικτική μέθοδο της ΛΠ και κατόπιν, αναλύοντας στο δεύτερο κεφάλαιο τις έννοιες των κανονικών μορφών, των μορφών Skolem καθώς και τις έννοιες της αντικατάστασης και της ενοποίησης, επεκτείναμε επίλυση σαν μέθοδο απόδειξης και στη ΛΚ. Τώρα θα ορίσουμε μια γλώσσα προγραμματισμού με την οποία θα κωδικοποιήσουμε και θα επεξεργαστούμε την καθημερινή πρακτική εμπειρία και τα γεγονότα του κόσμου μας, χρησιμοποιώντας κατά το δυνατό περισσότερα στοιχεία της μαθηματικής επιστήμης.

Τα κύρια στοιχεία αυτής της γλώσσας είναι ο τρόπος παρουσίασης των πληροφοριών, η λειτουργία του συμπερασματικού μηχανισμού της καθώς και η σχέση του μηχανισμού αυτού με την Μαθηματική Λογική.

Ο τρόπος παρουσίασης καθορίζεται από τους τύπους Horn, οι οποίοι όχι μόνο περιγράφουν απλά τα γεγονότα του κόσμου μας, αλλά και μετατρέπονται εύκολα σε κανονικές μορφές Skolem. Ο συμπερασματικός μηχανισμός της γλώσσας με την αντικατάσταση, ενοποίηση και δυαδική επίλυση, προσφέρει ένα αλγοριθμικό μαθηματικό εργαλείο για την διαχείριση δεδομένων και την εξαγωγή συμπερασμάτων από ένα σύνολο τύπων Horn.

Ξέροντας λοιπόν ότι η μέθοδος της δυαδικής επίλυσης εξετάζει τα δεδομένα ενός προβλήματος και εγγυάται πως αν υπάρχει λύση, αυτή η λύση

θα βρεθεί, είμαστε σε θέση να κάνουμε μια πρώτη διαισθητική παρουσίαση της Prolog με ένα παράδειγμα.

Θεωρούμε τα ακόλουθα δεδομένα εκφρασμένα σε τύπους Horn της ΛΚ:

$\pi_1$ : Άνθρωπος (Μάρκος) ←

$\pi_2$ : Άνθρωπος(Αυρήλιος) ←

$\pi_3$ : Θνητός( $x_1$ ) ← Άνθρωπος ( $x_1$ )

$\pi_4$ : Γεννήθηκε \_ Πομπηία (Αυρήλιος) ←

$\pi_5$ : Γεννήθηκε \_ Πομπηία (Μάρκος) ←

$\pi_6$ : Γέννηση (Αυρήλιος, 45) ←

$\pi_7$ : Γέννηση (Μάρκος, 40) ←

$\pi_8$ : Απεβίωσε ( $x_2$ , 79) ← Γεννήθηκε \_ Πομπηία \_ ( $x_2$ )

$\pi_9$ : Εξερράγη (ηφαίστειο, 79) ←

$\pi_{10}$ : Νεκρός ( $x_3$ ,  $t_2$ ) ← Θνητός ( $x_3$ ), Γέννηση ( $x_3$ ,  $t_1$ ),  $t_2 - t_1 > 150$

$\pi_{11}$ : Όχι \_ Ζωντανός( $x_4$ ,  $t_3$ ) ← Νεκρός ( $x_4$ ,  $t_3$ )

$\pi_{12}$ : Νεκρός ( $x_5$ ,  $t_4$ ) ← Όχι \_ Ζωντανός ( $x_5$ ,  $t_4$ )

$\pi_{13}$ : Νεκρός ( $x_6$ ,  $t_6$ ) ← Απεβίωσε ( $x_5$ ,  $t_5$ ),  $t_6 > t_5$

$\pi_{14}$ : ← Όχι \_ Ζωντανός ( $x$ , 1987)

Οι δεκατρείς πρώτοι τύποι Horn είναι οι πληροφορίες που δίνουμε στο πρόγραμμα. Η  $\pi_{14}$  είναι η ερώτηση που υποβάλλουμε:

“ποιος δεν είναι ζωντανός το έτος 1987;”

Η αντίστοιχη συνολοθεωρητική μορφή των προτάσεων  $\pi_1$ - $\pi_{14}$  είναι:

1. {Άνθρωπος(Μάρκος)}
2. {Άνθρωπος(Αυρήλιος)}
3. {Θνητός( $x_1$ ),  $\neg$  Άνθρωπος( $x_1$ )}
4. {Γεννήθηκε \_ Πομπηία(Αυρήλιος)}
5. {Γεννήθηκε \_ Πομπηία(Μάρκος)}
6. {Γέννηση(Αυρήλιος, 45)}

7. {Γέννηση(Μάρκος,40)}
8. {Απεβίωσε(x2,79),  $\neg$  Γεννήθηκε\_Πομπηία(x2)}
9. {Εξερράγη(ηφαίστειο,79)}
10. {Νεκρός(x3,t2),  $\neg$  Θνητός(x3),  $\neg$ Γέννηση(x3,t1),  $\neg$ (t2-t1)>150)}
11. {Όχι\_Ζωντανός(x4,t3).  $\neg$ Νεκρός(x4,t3)}
12. {Νεκρός(x5,t4,)  $\neg$  Όχι\_Ζωντανός(x5,t4)}
13. {Νεκρός(x6,t6)  $\neg$  Απεβίωσε (x6,t5),(t6>t5)}
14. {  $\neg$ Όχι\_Ζωντανός (x,1987)}.

Η αντίστοιχη παρουσίαση στη Prolog είναι:

- 1.άνθρωπος(μάρκος).
- 2.άνθρωπος(αυρήλιος).
- 3.θνητός(X1):-άνθρωπος (X1).
- 4.γεννήθηκε\_πομπηία(αυρήλιος).
- 5.γεννήθηκε\_πομπηία(μάρκος).
- 6.γέννηση(αυρήλιος,45).
- 7.γέννηση(μάρκος,40).
- 8.απεβίωσε(X2,79):-γεννήθηκε\_πομπηία(X2).
- 9.εξερράγη(ηφαίστειο,79).
- 10.νεκρός(X3,T2):-θνητός(X3), γέννηση(X3,T1),gt(T2-T1,150).
- 11.όχι\_ζωντανός(X4,T3):-νεκρός(X4,T3).
- 12.νεκρός(X5,T4):-όχι\_ζωντανός(X5,T4).
- 13.νεκρός(X6,T6):-απεβίωσε(X6,T5),gt(T6,T5).
- 14.?οχι\_ζωντανός(X,1987).

(όπου gt(X,Y) σημαίνει  $X > Y$ )

Η τελεία “.” χρησιμοποιείται στις περισσότερες εκδόσεις Prolog για να δηλωθεί το τέλος της εισαγωγής ενός τύπου Horn. Χρησιμοποιούμε φορμαλισμό της Prolog και γράφουμε μεταβλητές με κεφαλαία γράμματα και δείκτες πλάι στην αντίστοιχη μεταβλητή.

Η Prolog , όπως και η ΔΚ, προσπαθεί να αποδείξει το στόχο και την απάντησή της, αν στόχος αληθεύει, θα ενημερωθεί, δηλαδή θα δώσει τιμές, σε

όλες τις άγνωστες μεταβλητές που επαληθεύουν το στόχο. Ας εξετάσουμε πως γίνεται αυτό και τι σχέση έχει με την αντικατάσταση, την ενοποίηση και την επίλυση

### **Βήμα 1<sup>ο</sup>**

Ο στόχος μας είναι: **Όχι\_Ζωντανός(x,1987)**.

Η Prolog ψάχνει στα δεδομένα για έναν προγραμματικό τύπο Horn, που η κεφαλή του να ενοποιείται με το στόχο. Πράγματι, η αντικατάσταση  $\Theta_0 = \{x4/x, t3/1987\}$  ενοποιεί το στόχο με την κεφαλή του τύπου 11. Η Prolog αμέσως προσπαθεί να επιβεβαιώσει το σώμα του τύπου 11, θέτοντας σαν νέο στόχο, μια-μια και διαδοχικά όλες τις υποθέσεις αυτής της πρότασης.

Η δυαδική επίλυση στους 14 και 11 δίνει:

15. {Νεκρός(x,1987)} από 11 και 14.

### **Βήμα 2<sup>ο</sup>**

Ο στόχος μας τώρα είναι: **Νεκρός(x,1987)**

Η Prolog όπως και η επίλυση, προσπαθεί τώρα να ενοποιήσει το στόχο "Νεκρός(x,1987)" με κάποια κεφαλή προγραμματικού τύπου από τα δεδομένα της. Αυτό επιτυγχάνεται μέσω της αντικατάστασης  $\Theta_1 = \{x3/x, t2/1987\}$  στην πρόταση 10.

### **Βήμα 3<sup>ο</sup>**

Ο νέος στόχος για την Prolog είναι τώρα η τριάδα των υποθέσεων του τύπου 10 δηλαδή οι:

**Θνητός(x),**

**Γέννηση(x,t1)**

**gt(1987-t1,150)**

τις οποίες και προσπαθεί να ικανοποιήσει με την σειρά που παρουσιάζονται στον τύπο 10. Αντίστοιχα, η μέθοδος της επίλυσης θα μας έδινε τον τύπο 16:

16. {Θνητός(x),Γέννηση(x,t1),gt(1987-t1,150)}.

#### Βήμα 4<sup>ο</sup>

Η Prolog ενοποιεί τον τύπο “Θνητός(x) με την κεφαλή της πρότασης 3 χρησιμοποιώντας την αντικατάσταση  $\Theta_2 = \{x1/x\}$ . Όμοια, η επίλυση θα χρησιμοποιούσε την ίδια αντικατάσταση για να ενοποιήσει το “Θνητός(x)” με το “Θνητός(x1)” της πρότασης 3.

#### Βήμα 5<sup>ο</sup>

Ο νέος στόχος τώρα είναι: **Άνθρωπος(x)**, ο οποίος και ενοποιείται με την σειρά του μέσω της αντικατάστασης  $\Theta = \{x/Μάρκος\}$  με το γεγονός που περιγράφει ο τύπος 1.

Με την μέθοδο της επίλυσης θα είχαμε:

17. {¬Άνθρωπος (x), ¬ Γέννηση(x,t1), ¬ gt(1987-t1,150)}

από 3 και 10

18. {¬Γέννηση(Μάρκος,t1), ¬ gt(1987-t1,150)}

από 1 και 17

#### Βήμα 6<sup>ο</sup>

Η Prolog συνεχίζει την αποδεικτική πορεία της προσπαθώντας να ικανοποιήσει τον επόμενο στόχο από την τριάδα των στόχων του βήματος 3 ο οποίος τώρα είναι:

**Γέννηση(Μάρκος,t1).**

Η ενοποίηση επιτυγχάνει μέσω της αντικατάστασης  $\Theta_4 = \{t1/40\}$  με την κεφαλή του τύπου 7.

Η επίλυση αντίστοιχα θα συμπεράινε τον

19. {¬gt(1987-40,150)}

από 7 και 18

**Βήμα 7<sup>ο</sup>**

Ο τελευταίος προς ικανοποίηση στόχος από την τριάδα των στόχων του βήματος 3 είναι ο “gt(1987-40,150)” ο οποίος και ικανοποιείται από την Prolog διότι  $1987-40 > 150$ .

**Βήμα 8<sup>ο</sup>**

Ο στόχος του βήματος 2, “Νεκρός(x,1987)”, έχει ικανοποιηθεί με την ενημέρωση  $x = \text{Μάρκος}$  και συνεπώς ικανοποιήθηκε και ο στόχος του βήματος 1, “Όχι\_Ζωντανός(x,1987)”, με την ίδια ενημέρωση.

**Βήμα 9<sup>ο</sup>..**

Η Prolog δεν σταματά εδώ. Συνεχίζει προσπαθώντας να ενοποιήσει τον τελευταίο επιλεγμένο στόχο με διαφορετικό τρόπο, για βρει όλες τις δυνατές λύσεις. Αφού ικανοποιηθεί τελευταίος στόχος με όλους τους τρόπους και με την παραγωγή όλων των δυνατών λύσεων, η διαδικασία αυτή επαναλαμβάνεται για τον αμέσως προηγούμενο στόχο. Όταν αυτός ικανοποιηθεί με έναν διαφορετικό τρόπο, το πρόγραμμα συνεχίζει ικανοποιώντας πάλι τον αρχικά τελευταίο στόχο με τις αντικαταστάσεις που έγιναν τον προτελευταίο και ούτω καθ'εξής:

Ο τελευταίος στόχος “gt(1987-40,150)” δεν ενοποιείται με κάποιο διαφορετικό τρόπο, οπότε η Prolog προσπαθεί να ενοποιήσει, με διαφορετική αντικατάσταση αυτή την φορά, τον αμέσως προηγούμενο στόχο “Γεννήθηκε(Μάρκος,40)”. Με αυτή την μέθοδο ενοποιεί τελικά το “Θνητός(x)” μέσω της αντικατάστασης  $\Theta = \{x/\text{Αυρήλιος}\}$  με τον τύπο 2 και παράγει όπως και προηγουμένος από το βήμα 6, και για  $x = \text{Αυρήλιος}$  αυτή τη φορά, την νέα λύση “Όχι\_Ζωντανός(Αυρήλιος,1987)”.

Αυτός ο ευέλικτος μηχανισμός της Prolog που βρίσκει όλες τις δυνατές λύσεις λέγεται **επαναδρομηση** (backtracking) και θα αναλυθεί εκτενέστερα στην αντίστοιχη ενότητα.

## Α..Λογική και προγραμματισμός

Η διαφορά του λογικού προγραμματισμού και της γλώσσας Prolog από τον παραδοσιακό προγραμματισμό και γλώσσες όπως FORTRAN, BASIC, COBOL, PASCAL κτλ. βρίσκεται στις θεμελιώδεις αρχές του λογικού προγραμματισμού, τόσο στον σχεδιασμό, όσο και στην εκτέλεση ενός λογικού προγράμματος.

Ένα πρόγραμμα αποτελείται από δυο δομικά στοιχεία, την **Λογική** και τον **Έλεγχο** {Kow79,Llo87}. Με τον όρο Λογική χαρακτηρίζουμε όλες εκείνες τις σημαντικές έννοιες που προσδιορίζουν το ΤΙ κάνει ένα πρόγραμμα, ενώ με τον όρο Έλεγχο, όλες εκείνες τις συντακτικές έννοιες που προσδιορίζουν το ΠΩΣ το κάνει (π.χ. ο αλγόριθμος επίλυσης ενός προβλήματος). Αν θελήσουμε να περιγράψουμε όλα αυτά σε μία εξίσωση θα γράφαμε:

ΠΡΟΓΡΑΜΜΑ=ΛΟΓΙΚΗ+ΕΛΕΓΧΟ

Ένα πρόγραμμα γραμμένο σε γλώσσα BASIC ή σε κάποια άλλη γλώσσα παραδοσιακού προγραμματισμού αποτελείται από εντολές οι οποίες περιγράφουν ενέργειες που πρέπει να εκτελεσθούν βήμα προς βήμα από τον υπολογιστή για να έχει το πρόγραμμα το επιθυμητό αποτέλεσμα, π.χ. η εντολή σ'ένα πρόγραμμα BASIC

```
10 LET X=X+5
```

αυξάνει το περιεχόμενο της θέσης μνήμης που αντιστοιχεί στην μεταβλητή X κατά 5.

Οι γλώσσες προγραμματισμού όπως η BASIC χαρακτηρίζονται από **προστακτικές**(imperative) εντολές που περιγράφουν την **βήμα προς βήμα** συμπεριφορά του προγράμματος, έτσι ώστε, μετά από μια πεπερασμένη ακολουθία τέτοιων εντολών, να επιτυγχάνεται το σωστό και αναμενόμενο



αποτέλεσμα. Ο τρόπος δόμησης αυτών των εντολών κατά την διάρκεια σχεδιασμού του προγράμματος αποτελεί και το στοιχείο του ΕΛΕΓΧΟΥ.

Όμοια το στοιχείο της ΛΟΓΙΚΗΣ στην παραπάνω εντολή είναι η έκφραση “ $X+5$ ”, η οποία δεν αποτελεί εντολή αλλά ένα μικρό περιγραφικό πρόγραμμα, το οποίο περιγράφει άμεσα την αριθμητική τιμή που πρέπει να υπολογισθεί και έμμεσα μόνο το πώς θα υπολογισθεί αυτή η τιμή.

Έτσι η BASIC είναι μια γλώσσα “προστακτική” η οποία διαθέτει ένα “περιγραφικό” στοιχείο.

Αντίθετα, ένα πρόγραμμα Prolog θα μπορούσε να περιέχει την πρόταση:

Καλός(x):-Αγαπά(x,ανθρώπους)

η οποία είναι απλά ένας ορισμός της σχέσης μεταξύ των κατηγορημάτων “Καλός” και “Αγαπά”.

Ο σχεδιασμός λοιπόν ενός προγράμματος Prolog βασίζεται στη σωστή επιλογή των κατηγορημάτων που ορίζουν τις σχέσεις των αντικειμένων, έτσι ώστε να περιγράφεται πλήρως η σχέση των πληροφοριών εισόδου με τις πληροφορίες που περιμένουμε σαν αποτέλεσμα στην έξοδο.

Γενικά ένα πρόγραμμα σε μια παραδοσιακή γλώσσα προγραμματισμού, εκφράζει μια απεικόνιση από δεδομένα(input) στο αποτέλεσμα (output) του προγράμματος ενώ ένα πρόγραμμα σε μια γλώσσα Λογικού προγραμματισμού, εκφράζει μια σχέση μεταξύ των δεδομένων .Επειδή οι σχέσεις είναι γενικότερες από τις απεικονίσεις (οι σχέσεις ΔΕΝ είναι απαραίτητα μονοσήμαντες, ενώ οι απεικονίσεις είναι), οι δυνατότητες του Λογικού Προγραμματισμού είναι μεγαλύτερες από τις δυνατότητες του κλασικού προγραμματισμού.

Η επιλογή των κατηγορημάτων και των σχέσεων που εκφράζονται με αυτά τα κατηγορήματα αποτελούν την Prolog τη ΛΟΓΙΚΗ. Ο ΕΛΕΓΧΟΣ είναι αφ'ενός η σειρά με την οποία διατάσσουμε με τους τύπους και εφ'ετέρου μερικά δομικά στοιχεία ελέγχου, όπως η τομή(!),τα οποία προσφέρονται σαν συντακτικά αντικείμενα της Prolog . Η PROLOG είναι δηλαδή μια

**περιγραφική** γλώσσα η οποία διαθέτει κάποιο **προστακτικό** δομικό στοιχείο σαν στοιχείο ελέγχου.

Ας δούμε για παράδειγμα ένα πρόγραμμα που διαβάζει δυο αριθμούς και τυπώνει τον μεγαλύτερο. Για να γίνει σαφής η διαφορά του παραδοσιακού από τον Λογικό Προγραμματισμό θα δώσουμε πρώτα τα πρόγραμμα σε BASIC και στη συνέχεια το ίδιο ακριβώς πρόγραμμα σε Prolog:

### **Πρόγραμμα BASIC**

```
10 INPOUT "ARITHMOS1", X
20 INPUT "ARITHMOS2", Y
30 IF X > Y THEN 60
40 PRINT Y
50 GOTO 70
60 PRINT X
70 END
```

### **Πρόγραμμα Prolog**

```
program:-write("ARITHMOS1"),real(X),nl,
          write("ARITHMOS2"),real(Y),nl,
          greater(X,Y,Z),write(Z).

greater(X,X,X).
greater(X,Y,Y):-X < Y.
greater(X,Y,X):-Y < X.
?program.
```

Το "real",3.4.5., είναι ένα ειδικό κατηγορημα της Prolog που ελέγχει αν ένας αριθμός είναι πραγματικός και με το "nl" κάθε αριθμός γράφεται σε διαφορετική σειρά κατά την εκτύπωση.

Το πρόγραμμα σε BASIC είναι μια ακολουθία εντολών. Αυτές οι εντολές που εκτελούνται με την σειρά που υποδεικνύει το πρόγραμμα, αποτελούν τον έλεγχο, δηλαδή το σχεδιασμό και τη ροή του προγράμματος. Το στοιχείο της Λογικής σ' αυτό το πρόγραμμα BASIC βρίσκεται στη σχέση ">".

Αντίθετα το πρόγραμμα Prolog είναι ένα σύνολο από τύπους που περιγράφουν πλήρως την σχέση που καθορίζει την διάταξη δυο αριθμών, δηλαδή το κατηγορημα "μεγαλύτερο". Αυτό το σύνολο των τύπων εκφράζει τη Λογική, η οποία έχει και τον κυρίαρχο ρόλο σ' ένα πρόγραμμα Prolog, ενώ ο έλεγχος βρίσκεται στη σειρά που διατάσσουμε ή και ορίζουμε τα κατηγορήματα.

Ένα κλασσικό πρόγραμμα αποτελείται εν γένει από τα δεδομένα που θέλουμε να επεξεργαστούμε και από την ακολουθία των πράξεων που θέλουμε να γίνουν. Αφού τυποποιήσουμε τα δεδομένα του προβλήματος και τις διαδικασίες που θα μας δώσουν τα αποτελέσματα, η δομή ελέγχου καθορίζει τη σειρά με την οποία πρέπει να εκτελεστούν οι διάφορες διαδικασίες. Δηλαδή έχουμε μια σαφώς καθορισμένη ακολουθία διαδικασιών μερικές από τις οποίες είναι επαναληπτικές.

Η επιλογή του τυπικού που θα ακολουθήσουμε είναι καθοριστική, γιατί από αυτήν εξαρτάται ποια είδη επεξεργασίας μπορούν να αυτοματοποιηθούν. Την ευθύνη γι' αυτήν την επιλογή τη φέρει εξ ολοκλήρου ο προγραμματιστής.

Βασικό λοιπόν στοιχείο ενός κλασσικού προγράμματος είναι η ροή, η διάταξη των διαδικασιών, σύμφωνα με την σειρά κατά την οποία εκτελούνται αυτές οι διαδικασίες.

## B. Ο Λογικός Προγραμματισμός

Από τα μεγάλα προβλήματα των προγραμμάτων στον παραδοσιακό προγραμματισμό είναι το ότι χρειάζονται συνεχή αλλαγή και προσαρμογή σε νέες απαιτήσεις. Κάθε όμως από μέρους αλλαγή σ' ένα κλασσικό πρόγραμμα, επηρεάζει συνήθως τη συνολική του ροή. Για το λόγο αυτό, ο έλεγχος ενός έστω και ελάχιστα διαφοροποιημένου προγράμματος είναι επίπονη και χρονοβόρα διαδικασία.

Μία λύση στα προβλήματα του παραδοσιακού προγραμματισμού έδωσε ο Λογικός Προγραμματισμός μέσω της γλώσσας της Λ.Κ. Ένα πρόβλημα που πρέπει να λυθεί με τη βοήθεια του υπολογιστή, παρουσιάζεται σαν ένα σύνολο τύπων της ΛΚ. Από τους τύπους αυτούς, άλλοι εκφράζουν γεγονότα, π.χ. στο παράδειγμα ο τύπος “greater(X,X,X)”, άλλοι εκφράζουν κανόνες χειρισμού των γεγονότων, π.χ. στο ίδιο παράδειγμα ο τύπος “greater(X,Y,X):-X>Y”, και άλλοι εκφράζουν τα ερωτήματα που πρέπει να απαντηθούν και να δώσουν τη λύση του προβλήματος. Έτσι στο Λογικό Προγραμματισμό, και συγκεκριμένα στην Prolog, μετά την εφαρμογή των κανόνων, η λύση του προβλήματος θα είναι ή μία απάντηση των ερωτημάτων της μορφής “ναι” ή “όχι”, ή ένα σύνολο τιμών, οι οποίες θα αποτελούν τη ζητούμενη λύση.

Οι εντολές λοιπόν στον Λογικό Προγραμματισμό δεν έχουν αποκλειστικά λειτουργικό χαρακτήρα, όπως στον κλασσικό προγραμματισμό, αλλά είναι κατηγορήματα, στοιχεία της γλώσσας της ΛΚ, τα οποία είναι αληθή ή ψευδή, ανάλογα με την ερμηνεία των όρων τους. Η αλήθεια ή το ψεύδος συγκεκριμένων κατηγορημάτων ανάλογα με την ερμηνεία των μεταβλητών τους, μας δίνουν ακριβώς τις αντίστοιχες απαντήσεις της μορφής “ναι” ή “όχι” στα ερωτήματα του προγράμματος.

Οι απαντήσεις στο ερωτήματα που υποβάλλονται στο πρόγραμμα, δίνονται μόνον δε σχέση με τις πληροφορίες που έχουμε δώσει στο πρόγραμμα. Έτσι, αν ρωτήσουμε στο πρόγραμμα του παραδείγματος της Α. , αν το 4 είναι τέλειο τετράγωνο, δηλαδή τετράγωνο κάποιου φυσικού αριθμού,

το πρόγραμμα μην έχοντας δεδομένο το κατηγορημα που χαρακτηρίζει τα τέλεια τετράγωνα, θα μας απαντήσει “όχι”.

Δηλαδή, ο στόχος

?τέλειο\_τετράγωνο(4)

**αποτυγχάνει.** Εδώ πρέπει να δοθεί ιδιαίτερη προσοχή. Η αποτυχία της Prolog να επιβεβαιώσει το στόχο, δεν σημαίνει ότι πράγματι ο στόχος είναι ψευδής, σημαίνει μόνον ότι με τα δεδομένα του προγράμματος και με το διαθέσιμο μηχανισμό συμπερασμάτων, δεν μπορούμε να συμπεράνουμε ότι συγκεκριμένος στόχος είναι αληθής. Αυτή είναι και η **παραδοχή του κλειστού σύμπαντος** (Closed World Assumption),(Wat90):

Ένα κατηγορημα Q της ΛΚ θεωρείται από το πρόγραμμα ψευδές, αν το πρόγραμμα δεν μπορεί να αποδείξει ότι το Q είναι αληθές.

Έτσι, κάθε πρόγραμμα της Prolog εμφανίζεται σαν πλήρης περιγραφή του σύμπαντος και το σύμπαν χαρακτηρίζεται πλήρως από τα δεδομένα του προγράμματος. Ό,τι είναι σημαντικό στο σύμπαν, περιγράφεται στα δεδομένα και οι σχετικοί στόχοι επιτυγχάνουν, ενώ ό,τι δεν περιγράφεται στο πρόγραμμα, δεν είναι γνωστό, και οι σχετικοί στόχοι αποτυγχάνουν.

### Γ. Ιστορική εξέλιξη.

Μετά τον αλγόριθμο του Herbrand (1930), με τον οποίο μπορούμε να βρούμε μία ερμηνεία που να διαψεύδει έναν τύπο  $\varphi$  της ΛΚ, αν ο  $\varphi$  δεν είναι λογικά αληθής, ο Gilmore (1960) είναι ο πρώτος που προσπάθησε να εφαρμόσει τη μέθοδο του Herbrand σε υπολογιστή [CL73]. Ο Gilmore βασίστηκε στο ότι ένας τύπος είναι λογικά αληθής, αν και μόνον αν η άρνησή του διαψεύδεται σε κάποια ερμηνεία της γλώσσας. Έτσι, στο πρόγραμμά του υπάρχει μία διαδικασία παραγωγής προτάσεων της ΛΠ, οι οποίες ελέγχονται περιοδικά για ασυνέπεια.

Όμως το πρόγραμμα του Gilmore δεν ήταν ικανό να ανταπεξέλθει στις δυσκολίες που εμφανιζόταν κατά την επεξεργασία σύνθετων προτάσεων της ΛΚ. Μετά τη δημοσίευση του αλγόριθμου ενοποίησης του Robinson [Ro65], πρώτος ο Loveland το 1970 χρησιμοποίησε γραμμική επίλυση (linear resolution with selection function), επίλυση κατά την οποία επιλέγουμε έναν τύπο, επιλύουμε με έναν άλλον, την επιλύουσα την επιλύουμε με έναν τρίτο τύπο και συνεχίζουμε επιλύοντας πάντοτε την τελευταία επιλύουμε με έναν τρίτο τύπο και συνεχίζουμε επιλύοντας πάντοτε την τελευταία επιλύουσα μέχρι να βρούμε τον κενό προγραμματικό τύπο [CL73, LIo87].

Τα πρώτα βήματα στο λογικό προγραμματισμό οφείλονται στους Kowalski και Colmerauer [SS86] το 1972. Πρώτος ο Kowalski διατύπωσε τη διαδικαστική ερμηνεία της λογικής των τύπων Horn και έδειξε ότι ένας κανόνας της μορφής

$$A:-B_1, B_2, \dots, B_n$$

μπορεί και να διαβαστεί και να εκτελεστεί από μία αναδρομική γλώσσα προγραμματισμού.

Τον ίδιο καιρό ο Colmerauer και η ομάδα του κατασκεύασαν σε FORTRAN μία προγραμματική γλώσσα απόδειξης θεωρημάτων (theorem prover), η οποία υλοποιούσε τη διαδικαστική ερμηνεία του Kowalski. Ο θεμελιώδης λίθος της Prolog (PROgrammer en LOGique) είχε μπει!

Ο πρώτος διερμηνευτής (interpreter) της Prolog, το πρόγραμμα δηλαδή που μεταφράζει ένα κείμενο κατανοητό από τους ανθρώπους (source file) σε γλώσσα κατανοητή από μηχανή (machine code), γράφτηκε από τον Roussel με βάση τη θεωρητική δουλειά του Colmerauer στην Μασαλία το 1972 σε γλώσσα Algol.

Όμως το ερώτημα αν η λογική, και κατά συνέπεια όλα τα μαθηματικά, μπορούν να χρησιμοποιηθούν σαν γλώσσα προγραμματισμού, πρέπει να απαντηθεί αρνητικά [Wat90]: Ένα πρόβλημα διατυπωμένο σε μία γλώσσα προγραμματισμού, πρέπει να είναι επιλύσιμο μέσω ενός υπολογιστή, ενώ στη λογική και στα μαθηματικά υπάρχουν προβλήματα που ο υπολογιστής δεν είναι σε θέση να το λύσει, π.χ. το πρόβλημα της απόφασης για τη ΔΚ, ή το θεώρημα του Fermat για τα μαθηματικά.

Έτσι ο Λογικός Προγραμματισμός, δηλαδή οι γλώσσες προγραμματισμού που βασίζονται στη λογική, είναι πάντοτε περιορισμένες σ'ένα μέρος των τύπων της λογικής, π.χ. η Prolog ασχολείται μόνον με τύπους Horn, υποσύνολο των προτάσεων της ΔΚ.

Τα τελευταία χρόνια μελετούνται ιδιαίτερα οι σχέσεις μεταξύ διαφορετικών προγραμματικών γλωσσών, όπως ο **συναρτησιακός προγραμματισμός** (functional programming) δηλαδή προγραμματισμός με βάση τον λ-λογισμό [Th88], ο λογικός προγραμματισμός και ο προγραμματισμός τύπων αντικειμένων (object-oriented programming), προγραμματισμός βασισμένος στη θεωρία τύπων, κατά τον οποίο κάθε αντικείμενο αντιπροσωπεύεται από ένα σύνολο ιδιοτήτων τιμών και διαδικασιών [Th88]. Ιδιαίτερη προσπάθεια καταβάλλεται για να βρεθούν κατάλληλοι μετασχηματισμοί και ισοδυναμίες που να επιτρέπουν μετάβαση από τη μία προγραμματική γλώσσα στις άλλες, καθώς και ταυτόχρονη χρήση διαφορετικών φορμαλισμών στο ίδιο πρόγραμμα.

Οι περισσότερες γλώσσες Λογικού Προγραμματισμού, στην ουσία λογικά συστήματα απόδειξης θεωρημάτων με την μέθοδο της επίλυσης, είναι το πρώτο βήμα προς τον ιδεώδη Λογικό Προγραμματισμό.

Για τον ιδεώδη όμως προγραμματισμό θα πρέπει να λύσουμε πλήρως το πρόβλημα του Ελέγχου, δηλαδή:

I. Να έχουμε στην διαθεσή μας πιο ικανοποιητικές και πιο ευέλικτες διαδικασίες Ελέγχου σε κάθε γλώσσα Λογικού Προγραμματισμού.

II. Να μπορούμε να μεταβιβάσουμε τον "Έλεγχο κατά την διάρκεια του σχεδιασμού και της εκτέλεσης ενός προγράμματος αποκλειστικά και μόνο στον υπολογιστή.

Η λύση του προβλήματος του ελέγχου θα επιτρέψει στους προγραμματιστές και χρήστες του μέλλοντος να περιορίζουν την επικοινωνία τους με τον υπολογιστή στον πλήρη και σαφή ορισμό του προβλήματος, ενώ το πρόγραμμα με τη σειρά του θα αναλαμβάνει την επίλυση του προβλήματος και τον ολικό έλεγχο της ροής του προγράμματος.



## ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

### 1. Τα στοιχεία του προγράμματος

Ένα πρόγραμμα Prolog αποτελείται από τα **δεδομένα** και τα **ερωτήματα** στα οποία πρέπει να απαντήσει το πρόγραμμα. Δεδομένα και ερωτήματα είναι τύποι Horn.

Τα δεδομένα έχουν μια από τις ακόλουθες μορφές(\*) ή (\*\*), όπου:

$$A(c_1, \dots, c_x) \leftarrow \quad (*)$$

ή όπως γράφουμε συμβολικά στην Prolog

$$A(c_1, \dots, c_x).$$

όπου A κατηγορημα και  $c_1, \dots, c_x$  σταθερές της ΛΚ. Η τελεία δηλώνει το σημείο στο οποίο τελειώνει κάθε φορά ο αντίστοιχος δεδομένος τύπος.

$$A(a_1, \dots, a_x) \leftarrow B_1(b_1, \dots, b_\lambda), \dots, B_j(d_1, \dots, d_\mu) \quad (**)$$

ή όπως γράφουμε συμβολικά στην Prolog:

$$A(a_1, \dots, a_x) :- B_1(b_1, \dots, b_\lambda), \dots, B_j(d_1, \dots, d_\mu).$$

όπου A,  $B_1, \dots, B_j$  κατηγορήματα και  $a_1, \dots, a_x, b_1, \dots, b_\lambda, \dots, d_1, \dots, d_\mu$  όροι.

Τα δεδομένα της μορφής (\*) είναι **γεγονότα** (facts) του προγράμματος.

Δεδομένα της μορφής (\*\*) ονομάζονται **κανόνες** (rules) του προγράμματος. Οι κανόνες εκφράζουν σχέσεις μεταξύ των κατηγορημάτων των γεγονότων. Οι κανόνες συνεπαγωγικοί τύποι της ΛΚ στους οποίους εφαρμόζει το πρόγραμμα ενοποίηση και επίλυση, αποτελούν **το διαδικαστικό μέρος** (procedural part) του προγράμματος.

Το  $A(a_1, \dots, a_x)$  λέγεται **κεφαλή** και τα  $B_1(b_1, \dots, b_\lambda), \dots, B_j(d_1, \dots, d_\mu)$  λέγονται **ουρά** του κανόνα.

Τα γεγονότα και οι κανόνες αποτελούν τη **βάση δεδομένων** (database) του προγράμματος.

Τα ερωτήματα (queries) είναι οι τύποι στόχοι (goals), δηλαδή τύποι της μορφής

$$\leftarrow B_1(b_1, \dots, b_\lambda), \dots, B_1(d_1, \dots, d_\mu)$$

ή συμβολικά της Prolog:

$$? B_1(b_1, \dots, b_\lambda), \dots, B_1(d_1, \dots, d_\mu).$$

όπου  $B_1, \dots, B_j$  κατηγορήματα και  $a_1, \dots, a_\kappa, b_1, \dots, b_\lambda, \dots, d_1, \dots, d_\mu$  όροι της ΛΚ.

Η τελεία “.” μετά από κάθε γεγονός, κανόνα ή ερωτήματα δηλώνει το τέλος τους αντίστοιχου τύπου Horn.

Ας δούμε ένα πρόγραμμα Prolog με το ερώτημα “τι μπορεί να κλέψει ο Πέτρος”

κλέφτης(πέτρος).

αρέσει(μαρία, φαγητό).

αρέσει(μαρία, κρασί).

αρέσει(πέτρος, χρήματα).

αρέσει(πέτρος, X):-αρέσει(X, κρασί).

μπορεί\_να\_κλέψει(X, Y):-κλέφτης(X), αρέσει(X, Y).

? μπορεί\_να\_κλέψει(πέτρος, Y).

Τα γεγονότα είναι οι τέσσερις πρώτοι τύποι Horn, οι δύο επόμενοι τύποι είναι οι κανόνες, και ο τελευταίος τύπος είναι ο στόχος (το ερώτημα) του προγράμματος. Η Prolog θα μας απαντήσει

Y=χρήματα

Y= μαρία

δηλαδή ο Πέτρος μπορεί να κλέψει χρήματα και την Μαρία.

Έτσι το πρόγραμμα Prolog είναι ένα σύνολο γεγονότων, κανόνων και ερωτήσεων, χωρίς ιδιαίτερες οδηγίες για την ροή και τον έλεγχο του προγράμματος. Αυτή ακριβώς η δομή δείχνει και τη μεγάλη ελαστικότητα της Prolog: προσθήκες και βελτιώσεις στο πρόγραμμα γίνονται με αναγραφή ή διαγραφή γεγονότων και κανόνων μέσω των εντολών assert και retract.

Τα γεγονότα σε ένα πρόγραμμα Prolog μπορούν να ερμηνευθούν σαν δηλώσεις (declarations) σχετικά με το σύμπαν του προγράμματος, δηλαδή αποτελούν ισχυρισμούς οι οποίοι περιγράφουν έναν συγκεκριμένο κόσμο. Οι κανόνες μπορούν να ερμηνευθούν όχι μόνον σαν δηλώσεις, αλλά και σαν

διαδικασίες (*procedures*) του προγράμματος. Οι παραδοσιακές γλώσσες προγραμματισμού επιδέχονται ουσιαστικά *διαδικαστική ερμηνεία* (*procedural interpretation*), ενώ η Prolog είναι ίσως η μόνη γλώσσα προγραμματισμού που επιδέχεται και *διαδικαστική ερμηνεία* και *δηλωτική ερμηνεία* (*declarative interpretation*)

## 2. Τα γεγονότα

Ένα γεγονός είναι μια σχέση, δηλαδή ένα κατηγορημα, μεταξύ ορισμένων αντικειμένων, τα οποία είναι σταθερές της γλώσσας ΛΚ.

Τα γεγονότα της Prolog εκφράζουν ακριβώς αντίστοιχα γεγονότα της καθομιλουμένης γλώσσας:

<b>καθομιλουμένη</b>	<b>Prolog</b>
το καναρίνι είναι πουλί	πουλί(καναρίνι)
Ο Νίκος είναι άνδρας	άνδρας(νίκος)
Η Μαρία αρέσει στον Πέτρο	αρέσει (πέτρος,μαρία)
Ο Πέτρος μπορεί να κλέψει την Μαρία	μπορεί_να_κλέψει((πέτρος,μαρία)
Οι γονείς της Όλγας είναι ο Κώστας	γονείς(όλγα,κώστας, ελένη)
και η Ελένη.	

Τα ονόματα των κατηγορημάτων καθώς και των σταθερών στα κατηγορήματα, αρχίζουν πάντοτε με μικρό γράμμα. Οι λέξεις σε κατηγορήματα που εκφράζονται με παραπάνω από μια λέξη, συνδέονται με κάτω παύλα”\_”. Οι τελείες δηλώνουν το τέλος κάθε γεγονότος.

Η θέση των σταθερών στα κατηγορήματα είναι διατεταγμένη. Το γεγονός “αρέσει(μαρία,πέτρος)”, δηλώνει ότι ο Πέτρος αρέσει στην Μαρία και είναι διαφορετικό από το γεγονός “αρέσει(πέτρος, μαρία)”.

Η επιλογή των κατηγορημάτων και των ονομάτων που θα χρησιμοποιηθούν, γίνεται από τον προγραμματιστή. Σημασία έχει το πως ερμηνεύει εκείνος τα γεγονότα. Π.χ. το γεγονός “πατέρας(νίκος, χρήστος)” μπορεί να ερμηνευθεί σαν “ ο Νίκος είναι πατέρας του Χρήστου” ή σαν “ο Χρήστος είναι πατέρας του Νίκου”. Μπορούμε να επιλέξουμε ανάμεσα στις δύο ερμηνείες, αρκεί να μην επιλέξουμε και τις δύο!! Δεν πρέπει να περιμένουμε σωστά αποτελέσματα από ένα πρόγραμμα που περιέχει τα γεγονότα

πατέρας(νίκος,χρήστος).

πατέρας(γιώργος,κώστας).

και το πρώτο γεγονός να ερμηνευθεί στην καθομιλουμένη σαν

“ο πατέρας του Νίκου είναι ο Χρήστος”

και το δεύτερο γεγονός σαν

“ο πατέρας του Κώστα είναι ο Γιώργος”

### 3.Οι κανόνες

Ένας κανόνας της Prolog είναι μια συνεπαγωγική πρόταση της ΛΚ. Οι κανόνες εκφράζουν πληροφορίες και σχέσεις γενικότερες και πιο σύνθετες από τις πληροφορίες που εκφράζουν τα γεγονότα.Π.χ.

πατέρας(χρήστος,γιώργος).

γιός(κώστας,γιάννης).

πατέρας(X,Y):-γιός(Y,X)

Οι δυο πρώτες προτάσεις είναι γεγονότα. Δηλώνουν τις σχέσεις των συγκεκριμένων ατόμων, Χρήστου-Γιώργου, και Κώστα-Γιάννη αντίστοιχα. Η τρίτη πρόταση εκφράζει ένα γενικό συμπερασματικό κανόνα σχετικά με τα κατηγορήματα των δεδομένων: αν ο Y είναι ο γιός του X, τότε ο X είναι πατέρας του Y.

Οι μεταβλητές στους κανόνες, όπως και στα ερωτήματα, γράφονται με κεφαλαία γράμματα. Η κεφαλή χωρίζεται από την ουρά με το σύμβολο”:-

”(neck symbol). Αν η ουρά του κανόνα αποτελείτε από περισσότερα του ενός κατηγορήματα, αυτά τα κατηγορήματα χωρίζονται μεταξύ τους με κόμμα”,.”.

Ας δούμε ένα παράδειγμα:

αρέσει(κώστας,παγωτό).

αρέσει(κώστας,άννα).

αρέσει(κώστας,τάβλι).

φαγητό(παγωτό).

τρώει(X,Y):-φαγητό(Y), αρέσει(X,Y).

Εδώ η ουρά του κανόνα αποτελείτε από δυο κατηγορήματα, τα “φαγητό(Y)” και “αρέσει(X,Y)”. Δηλαδή, αυτός ο κανόνας δηλώνει ότι για όλα τα X,Y, Αν το Y είναι φαγητό και αν το X αρέσει το Y, τότε ο X τρώει το Y.

Στην Prolog έχουμε την δυνατότητα να γράψουμε δύο ή και περισσότερους κανόνες με την ίδια κεφαλή σαν ένα νέο κανόνα, με κεφαλή του νέου κανόνα την κοινή κεφαλή, χρησιμοποιώντας το σύμβολο”,.”. Π.χ. οι κανόνες

αγαπάει(X,Y):-αρέσει(Y,X).

αγαπάει(X,Y):-καλός(Y).

δηλώνονται και σαν ένας κανόνας:

αγαπάει(X,Y):-αρέσει(Y,X); καλός(Y).

και ερμηνεία “ ο X αγαπάει τον Y, αν ο Y αρέσει στον Y ή ο Y είναι καλός”.

Η Prolog αριθμεί τα γεγονότα και τους κανόνες κάθε βάσης δεδομένων. Έτσι π.χ. στην τελευταία βάση δεδομένων ο “αρέσει(κώστας,παγωτό)” είναι ο πρώτος τύπος Horn της βάσης και ο

“τρώει(X,Y):-φαγητό(Y),αρέσει(X,Y)”

είναι ο πέμπτος.

## 4. Τα ερωτήματα

Τα **ερωτήματα** στην Prolog, τύποι, στόχοι είναι τύποι σχετικά με τους όρους και τα κατηγορήματα της ΛΚ που εμφανίζονται στα δεδομένα του προγράμματος. Π.χ. για τα δεδομένα του προηγούμενου παραδείγματος μπορούμε να ρωτήσουμε:

?αρέσει(κώστας,X).

?αρέσει(X,Y),φαγητό(Y).

και η Prolog να απαντήσει στην πρώτη ερώτηση με

X=παγωτό

X=άννα

X=τάβλι

και στην δεύτερη ερώτηση με

X=κώστας

Y=παγωτό

Τα ερωτήματα αρχίζουν με "?", ερωτηματικό, και τελειώνουν με τελεία. Ένα ερώτημα, π.χ. "?αρέσει(κώστας,X).", ενδέχεται να έχει πολλές απαντήσεις. Ένα σύνθετο ερώτημα π.χ. "?αρέσει(X,Y),φαγητό(Y).", έχει πολλούς υποστόχους, οι οποίοι χωρίζονται μεταξύ τους με κόμμα. Στα σύνθετα ερωτήματα ίδιες μεταβλητές αναφέρονται πάντοτε στον ίδιο όρο. Π.χ. ρωτήσαμε "?αρέσει(X,Y),φαγητό(Y).", δηλαδή ρωτήσαμε από το πρόγραμμα να βρει το X και εκείνο το Y που αρέσει στο X και είναι φαγητό. Το ερώτημα "?αρέσει(X,Y),φαγητό(X)." θα αποτύγχανε, δηλαδή η Prolog δεν θα μπορούσε να βρει μια κατάλληλη τιμή για το X και θα απαντούσε "όχι".

Ας δούμε για παράδειγμα την περιγραφή ενός πάρτυ:

αρέσει(κώστας,ελένη).

αρέσει(γιάννης,μαρία).

αρέσει(φαίδρα,νίκος).

αρέσει(καίτη,νίκος).

ακούει(γιάννης,ροκ).

- ακούει(σπύρος,ρέγκε).
- ακούει(άννα, χέβι-μέταλ).
- πίνει(κώστας,βότκα).
- πίνει(γιάννης,τζιν).
- πίνει(καίτη,κρασί).
- πίνει(φαίδρα,κρασί).
- πίνει(άννα,κόκα-κόλα).
- καπνίζει(κώστας).
- καπνίζει(καίτη).
- ζευγάρι(άννα,X):-ακούει(X,ροκ).
- ζευγάρι(νίκος,X):-πίνει(X,κρασί),αρέσει(X,νίκος),καπνίζει(X).
- ?αρέσει(γιάννης,μαρία). (1)
- ?αρέσει(στέλιος,μαρία). (2)
- ?πίνει(φαίδρα,κρασί),ακούει(φαίδρα,ροκ). (3)
- ?ζευγάρι(νίκος,X). (4)
- ?ακούει(X,Y). (5)
- ?αρέσει(καίτη,X),αρέσει(φαίδρα,X). (6)
- ?αρέσει(X,Y),ακούει(X,ροκ),πίνει(Y,βότκα). (7)

Ερωτήματα όπως (1),(2) και(3) είναι **ερωτήματα επιβεβαίωσης των δεδομένων**, ενώ ερωτήματα της μορφής(4) έως (7) είναι **ερωτήματα διαχείρισης των δεδομένων**. Τα (3),(6), και(7) είναι **σύνθετα συζευκτικά ερωτήματα**. Η Prolog θα μας δώσει τις εξής απαντήσεις:

- (1) ναι
- (2) όχι
- (3)όχι
- (4) X=καίτη
- (5) X=γιάννης  
Y=ροκ  
X=σπύρος

$Y = \text{ρέγκε}$

$X = \text{άννα}$

$Y = \text{χέβι μέταλ}$

(6)  $X = \text{νίκος}$

(7) όχι

Στα ερωτήματα (2),(3 και(7) το πρόγραμμα Prolog απέτυχε.

Για την ερώτηση (2), δεν υπάρχει πληροφορία σχετική με τον Στελιο και τη Μαρία στα δεδομένα του προγράμματος.

Για την ερώτηση (3), ο πρώτος υπόστοχος επιτυγχάνει, η Φαίδρα πίνει κρασί, αλλά δεν υπάρχει πληροφορία στα δεδομένα σχετικά με το τι μουσική ακούει η Φαίδρα. Έτσι ο δεύτερος υπόστοχος, και κατά συνέπεια (γιατί;) και ο στόχος (3) αποτυγχάνει.

Η ερώτηση (7) αποτυγχάνει, γιατί στα δεδομένα του πάρτυ δεν υπάρχουν τιμές για τα  $X$  και  $Y$ , οι οποίες να ικανοποιούν ταυτόχρονα (γιατί)και τους τρεις υποστόχους.



## ΣΥΝΤΑΞΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

### Τα αντικείμενα της Prolog

Όπως έχουμε ήδη πει, η γλώσσα που χρησιμοποιούμε στην Prolog είναι ένα υποσύνολο της γλώσσας της ΛΚ, εμπλουτισμένο με σύμβολα τα οποία εξυπηρετούν συγκεκριμένες ανάγκες. Έτσι οι όροι της Prolog είναι μεταβλητές, σταθερές και συναρτήσεις επί μεταβλητών και σταθερών. Οι σταθερές μπορεί να είναι ατομικοί όροι, αριθμοί, ή ακολουθίες ειδικών χαρακτήρων. Τα βασικά αντικείμενα της γλώσσας της Prolog είναι τα κατηγορήματα και οι λίστες (lists). Στην συνέχεια θα εξετάσουμε αναλυτικά τους όρους και τα βασικά αντικείμενα της Prolog.

### Το αλφάβητο της Prolog

Τα περισσότερα από τα παραδείγματα προγραμμάτων Prolog που είδαμε, είναι γραμμένα με ελληνικό αλφάβητο για καλύτερη κατανόησή τους. Γενικά όμως, στις εκδόσεις Prolog που υπάρχουν στην αγορά, χρησιμοποιείται το αγγλικό αλφάβητο. Πληροφορίες για ελληνικές εκδόσεις της Prolog υπάρχουν στο [Ξαν90].

Έτσι το αλφάβητο της Prolog αποτελείται από:

- I. Τα κεφαλαία και μικρά γράμματα του αγγλικού αλφαβήτου:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

- ii. Τα ψηφία 0123456789

- iii. Σύμβολα όπως !#\$%&\*()\_=-+<>?/.,!

ανάλογα με την εκάστοτε διάλεκτο της Prolog που χρησιμοποιούμε.

- iv. Χαρακτήρες που εκτελούν λειτουργίες χωρίς να εμφανίζονται άμεσα στην οθόνη, π.χ. κενό ή νέα γραμμή.

## Οι μεταβλητές

Οι μεταβλητές στην Prolog είναι γράμματα ή λέξεις γραμμένες με γράμματα του αλφάβητου της Prolog. Οι μεταβλητές αρχίζουν πάντοτε με κεφαλαίο γράμμα ή με κάτω παύλα. Π.χ. μεταβλητές είναι τα

X,Y, Αποτέλεσμα, Apotelesma, Father, Who, \_μαρία, A3

ενώ τα

χ,y, αποτελέσματα, apotelesma, μαρία

ΔΕΝ είναι μεταβλητές. Επίσης, δεν θεωρούνται μεταβλητές τα

1ος\_αθλητής (αρχίζει με αριθμό)

“Αποτέλεσμα” (είναι κλεισμένο σε εισαγωγικά)

3579 . (είναι γραμμένο με αριθμούς)

Έτσι, αν γράψουμε

γυναίκα (Μητέρα) (1)

γυναίκα (μητέρα) (2)

το γεγονός (1) δηλώνει στην Prolog ότι κάθε σταθερά που υπάρχει στο πρόγραμμα είναι γυναίκα (γιατί;), ενώ το γεγονός (2) δηλώνει ότι η συγκεκριμένη σταθερά “μητέρα” είναι γυναίκα.

Η Prolog προσφέρει μία ακόμη δυνατότητα, τη χρήση ανώνυμων μεταβλητών, δηλαδή μεταβλητών των οποίων δεν χρειαζόμαστε το όνομα και δηλώνουμε μόνον τη θέση τους με κάτω παύλα. Π.χ. γράφοντας στην Prolog

μητέρα(ελένη,\_)

δηλώνουμε ότι η Ελένη είναι μητέρα κάποιου προσώπου το οποίο δεν έχει σημασία για το υπόλοιπο πρόγραμμα, ή ρωτώντας

?γονείς(νίκος,\_Μητέρα).

ενδιαφερόμαστε μόνο για τη μητέρα του Νίκου και δηλώνουμε τον πατέρα του Νίκου με ανώνυμη μεταβλητή. Βασικά χρησιμοποιούμε ανώνυμη μεταβλητή, όταν αυτή η συγκεκριμένη μεταβλητή εμφανίζεται μόνο μία φορά στο πρόγραμμα. Όπως θα δούμε στην σχετική ενότητα, με τη χρήση

ανώνυμων μεταβλητών, απλουστεύουμε τη διαδικασία ενοποίησης και κάνουμε οικονομία χρόνου.

## Οι σταθερές

Οι σταθερές ακολουθίες (string) συμβόλων της Prolog, χωρίζονται σε τρεις κατηγορίες, ανάλογα με τη μορφή τους [Bra88]. Έτσι, οι σταθερές της Prolog μπορεί να είναι

- I Ατομικοί όροι
- ii. Αριθμοί
- iii. Ακολουθίες ειδικών χαρακτήρων

Ένας ατομικός όρος είναι μια ακολουθία αλφαριθμητικών χαρακτήρων και “-”, αποτελείται δηλαδή από γράμματα, αριθμούς και ενδεχομένως κάτω παύλες στο εσωτερικό του (η κάτω παύλα στην αρχή της ακολουθίας δηλώνει μεταβλητή). Το αρχικό σύμβολο κάθε ατομικού όρου είναι πάντοτε μικρό γράμμα. Π.χ.

y342, ελένη, tom, κόκκινο\_μήλο, work\_procedure

Δεν είναι ατομικοί όροι π.χ. τα

Πέτρος	αρχίζει με μικρό γράμμα
_x5	αρχίζει με κάτω παύλα
3t	αρχίζει με αριθμό
α3.2	έχει τελεία.

Ατομικός όρος επίσης θεωρείται και κάθε ακολουθία συμβόλων που είναι κλεισμένη σε απλά εισαγωγικά. Π.χ. τα

‘x’, ‘Πέτρος’, ‘5tx’, ‘Ο Γιάννης με την ωραία γραβάτα’  
είναι ατομικοί όροι.

Οι αριθμοί [Bra88] είναι ακέραιοι ή πραγματικοί, π.χ.

211345, -32, 0.0000013, -5.7

Αρχικά η Prolog κατασκευάστηκε σαν γλώσσα λογικών υπολογισμών και η ικανότητά της να κάνει αριθμητικούς υπολογισμούς ήταν σχετικά μικρή. Η τελευταία όμως έκδοση, Prolog III [Col90], έχει μεγάλες δυνατότητες για αριθμητικούς υπολογισμούς.

Με τις ακολουθίες ειδικών χαρακτήρων της Prolog κατασκευάζουμε σύμβολα στα οποία αποδίδουμε μια συγκεκριμένη σημασία, π.χ. τα

====> ή ---->

που μπορούν να χρησιμοποιηθούν σαν σύμβολα για τη συνεπαγωγή,

## Τα κατηγορήματα

Τα κατηγορήματα, εκφράζουν σχέσεις και ιδιότητες όρων. Π.χ. το κατηγορήμα

μεγαλύτερο(+ (X,2),:(Y,3))

δηλώνει ότι ο όρος “+(X,2)” ή “X+2”, είναι μεγαλύτερος από τον όρο “:(Y,3)” ή “Y:3”. Τα “+” και “:” είναι σύμβολα συναρτήσεων της πρόσθεσης και της διαίρεσης αντίστοιχα, τα X, Y είναι μεταβλητές και τα 2, 3 σταθερές. Τα κατηγορήματα στην Prolog αρχίζουν συνήθως με μικρό γράμμα.

Στην Prolog έχουμε τη δυνατότητα να κατασκευάσουμε σύνθετα κατηγορήματα ή δομές (structures) από κατηγορήματα που ήδη υπάρχουν στο πρόγραμμα. Π.χ.. ας πάρουμε τα γεγονότα που περιγράφουν μία οικογένεια:

πατέρας(κώστας).

μητέρα(ελένη).

παιδί(όλγα).

παιδί(σπύρος).

Τα “πατέρας”, “μητέρα” και “παιδί” είναι κατηγορήματα και τα “κώστας”, “ελένη”, “όλγα”, “σπύρος” είναι σταθερές.

Τα δεδομένα (\*) μπορούν να εκφραστούν με ένα μόνον κατηγορήμα: οικογένεια(πατέρας(κώστας), μητέρα(ελένη), παιδιά(όλγα, σπύρος)) (\*\*).

Στην έκφραση (\*\*), “οικογένεια” είναι το κατηγορημα, τα “πατέρας”, “μητέρα” και “παιδιά” δεν είναι πια κατηγορήματα αλλά συναρτήσεις [th88]. Δηλαδή, τα “πατέρας(κώστας)” ,”μητέρα(ελένη)” και “παιδιά(όλγα, σπύρος)” στην (\*\*),σαν συγκεκριμένες τιμές συναρτήσεων, θεωρούνται σταθερές. Έτσι, το “πατέρας (κώστας)” στα γεγονότα (\*) είναι διαφορετικό συντακτικό αντικείμενο από το “πατέρας(κώστας)” στο (\*\*).

Με τον ίδιο τρόπο, το γεγονός

αρέσει(γιώργος, παγωτό).

όπου το “αρέσει” είναι κατηγορημα και τα “γιώργος” και “παγωτό” είναι σταθερές, εκφράζεται επίσης και από τα κατηγορήματα

παγωτό (αρέσει(γιώργος)). (1)

ή γιώργος(αρέσει(παγωτό)). (2)

όπου στο (1) “παγωτό”, “αρέσει” και “γιώργος” είναι αντίστοιχα κατηγορημα, συνάρτηση και σταθερά, ενώ στο (2) τα “γιώργος”, “αρέσει” και “παγωτό” είναι αντίστοιχα κατηγορημα, συνάρτηση και σταθερά.

Η Prolog διακρίνει αν μια συγκεκριμένη έκφραση είναι σταθερά συνάρτηση ή κατηγορημα, ανάλογα με τη θέση της έκφρασης σ'ένα γεγονός ή σ'έναν κανόνα ή σ'ένα ερώτημα. Αυτός ο τρόπος διάκρισης θα φανεί καλύτερα στην επόμενη παράγραφο, όπου θα αναπαραστήσουμε τα κατηγορήματα σαν δένδρα.

Το μεγάλο πλεονέκτημα της χρήσης σύνθετων κατηγορημάτων φαίνεται και στο παράδειγμα της οικογένειας (\*) και (\*). Τα σύνθετα κατηγορήματα προσφέρουν συνοπτική έκφραση και ευέλικτο χειρισμό των δεδομένων. Π.χ., αν ενδιαφερόμαστε για τους γονείς της οικογένειας, τα αντίστοιχα ερωτήματα για το πρόγραμμα (\*) θα είναι:

? πατέρας(X).

?μητέρα(Y).

ενώ με τη χρήση σύνθετου κατηγορηματος στο (\*\*) θα έχουμε ένα μόνον ερώτημα:

? οικογένεια[πατέρας(X), μητέρα(Y),\_].

όπου η ανώνυμη μεταβλητή, δηλώνει ότι δεν ενδιαφερόμαστε για τις τιμές της συνάρτησης “παιδιά”.

## ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ ΚΑΙ Η ΔΙΑΔΙΚΑΣΙΑ ΕΠΑΝΑΔΡΟΜΗΣΗΣ

Όπως έχουμε ήδη αναφέρει, ένα Prolog πρόγραμμα έχει δύο ερμηνείες: τη λογική ή δηλωτική (declarative) και τη διαδικαστική (procedural). Σ' αυτή την ενότητα θα εξετάσουμε τη διαδικαστική ερμηνεία της Prolog, δηλαδή, θα περιγράψουμε ΠΩΣ η Prolog εξάγει συμπεράσματα.

Έχουμε ήδη παρουσιάσει τη διαδικασία ενοποίησης και τη μέθοδο της επίλυσης. Εφαρμόζοντας τη διαδικασία ενοποίησης, καταλήγουμε σ' ένα σύνολο βασικών όρων του προγράμματος. Στο επόμενο βήμα, εφαρμόζουμε επαναληπτικά επίλυση στο σύνολο βασικών όρων καταλήγοντας έτσι σε συμπεράσματα μέσω της απόδειξης του κενού προγραμματικού τύπου. Όμως, ενώ με τη μέθοδο της επίλυσης είμαστε σίγουροι ότι οποιαδήποτε πρόταση που μπορεί να πάρει αληθοτιμή, θα την πάρει, στην Prolog δεν μπορούμε να κατευθύνουμε την εξαγωγή συμπερασμάτων στις προτάσεις που μας ενδιαφέρουν, πράγμα που έχει μεγάλη σημασία για την αποδοτικότητα (efficiency) των προγραμμάτων.

Π.χ., ας πάρουμε τις προτάσεις της ΛΠ:

1. D
2. E
3.  $BV \rightarrow E$
4.  $AV \rightarrow D$
5.  $AV \rightarrow B$

και ας υποθέσουμε ότι τίθεται το πρόβλημα αν η A ισχύει. Εισάγουμε την πρόταση

6.  $\neg A$

και έχουμε την ακόλουθη απόδειξη μέσω επίλυσης της A:

7. B από 2. και 3.  
 8. A από 7. και 5.  
 9. □ από 8. και 6. (\*)

Όμως η A μπορεί να αποδειχθεί και με την ακόλουθη απόδειξη με επίλυση:

- 7'. A από 1. και 4.  
 8'. □ από 7' και 6. (\*\*)

Η απόδειξη (\*) περιέχει περισσότερα βήματα από την (\*\*), και η εφαρμογή της απαιτεί περισσότερο χρόνο. Επίσης, στην απόδειξη (\*) χρειάστηκε να συμπεράνουμε την πρόταση B, η οποία δεν έχει σχέση με την A.

Η Prolog ενσωματώνει στο συμπερασματικό της μηχανισμό μια ειδική διαδικασία έρευνας του σύμπαντος των όρων ενός προγράμματος. Από τη μία πλευρά, η διαδικασία αυτή επιτρέπει στον προγραμματιστή να ελέγχει την ακολουθία των υπολογισμών, από την άλλη εγγυάται την εύρεση όλων των δυνατών λύσεων ενός προβλήματος [Llo87]. Η διαδικασία αυτή ονομάζεται επαναδρομή (backtracking).

Π.χ., έστω η ακόλουθη Prolog βάση δεδομένων:

- κλέφτης(γιάννης). /\*1\*/  
 αρέσει(μαρία, φαγητό). /\*2\*/  
 αρέσει(μαρία, κρασί). /\*3\*/  
 αρέσει(γιάννης, X) :- αρέσει(X, κρασί). /\*4\*/  
 μπορεί\_να\_κλέψει(X, Y) :- κλέφτης(X),  
 αρέσει(X, Y). /\*5\*/

(Στις περισσότερες εκδόσεις της Prolog χρησιμοποιούμε τις ακολουθίες χαρακτήρων “/\*” για να εισάγουμε και “\*/” για να τελειώνουμε σχόλια, μη εκτελέσιμα από το πρόγραμμα. Έτσι στο παραπάνω πρόγραμμα έχουμε εισάγει την αρίθμηση των προγραμματικών τύπων σαν σχόλια.)

Ας υποθέσουμε τώρα ότι εισάγουμε το ερώτημα στόχο:

- ? μπορεί\_να\_κλέψει(γιάννης, Z).



/\* Τι μπορεί να κλέψει ο Γιάννης \*/

Τα βήματα που ακολουθούνται για την ικανοποίηση του στόχου είναι:

## Βήμα 1

? μπορεί\_να\_κλέψει(γιάννης, Z).

Η βάση δεδομένων ερευνάται για να ενοποιηθεί ο στόχος μ'ένα γεγονός ή κάποια κεφαλή κανόνα.

X/γιάννης

Η ενοποίηση επιτυγχάνει με την κεφαλή του κανόνα

Z/Y

5 και μέσω των αντικαταστάσεων X/γιάννης, Z/Y.

Η θέση του κανόνα 5 της βάσης δεδομένων καταγράφεται.

5

Οι μεταβλητές που εμφανίζονται στους προγραμματικούς τύπους του σώματος του κανόνα 5 ενημερώνονται επίσης μέσω των ίδιων αντικαταστάσεων.

κλέφτης(γιάννης)

αρέσει(γιάννης, Y)

## Βήμα 2

? κλέφτης(γιάννης).

Οι υποστόχοι του κανόνα θέτονται ένας κάθε φορά προς ικανοποίηση.

Η βάση δεδομένων ερευνάται από την κορυφή, προγραμματικός τύπος 1, για την ενοποίηση του

yes

1 υποστόχου. Η ενοποίηση επιτυγχάνει με τον προγραμματικό τύπο 1. Ο τύπος αυτός δεν περιέχει μεταβλητές, έτσι ο υποστόχος ικανοποιείται.

**Βήμα 3**

? αρέσει(γιάννης, Y).

X/Y

Ο δεύτερος υποστόχος τίθεται για ικανοποίηση. Η βάση δεδομένων ερευνάται από την κορυφή για την ενοποίηση του υποστόχου. Η ενοποίηση επιτυγχάνει με τον κανόνα 4 και μέσω της αντικατάστασης X/Y.

αρέσει(Y, κρασί) 4

Οι μεταβλητές του προγραμματικού τύπου του σώματος του κανόνα 4 ενημερώνονται επίσης.

**Βήμα 4**

? αρέσει(Y, κρασί).

Ο προγραμματικός τύπος του σώματος του κανόνα 4 τίθεται για ικανοποίηση. Η βάση δεδομένων ερευνάται και πάλι από την κορυφή για την ενοποίηση του στόχου.

**Βήμα 5**

? μπορεί\_να\_κλέψει(γιάννης,Z).

Y/μαρία Η προσπάθεια ενοποίησης με τον 2 αποτυγχάνει γιατί “φαγητό” και “κρασί” δεν ενοποιούνται.

Η θέση του προγραμματικού τύπου με τον οποίο έγινε η προσπάθεια ενοποίησης, 2, καταγράφεται.

**Βήμα 6**

Η Prolog προσπαθεί να ικανοποιήσει τον στόχο με άλλον τρόπο και προχωρεί

2	σ'επόμενους προογρ. τύπους
3	του προογράμματος. Η
Υ/μαρία	ενοποίηση επιτυγχάνει με τον 3.

## Βήμα 7

Ο αρχικός στόχος μπορεί να κλείψει (γιαάννης, Z) έχει ικανοποιηθεί μέσω της αντικατάστασης Z/μαρία. Η Prolog προσπαθεί τώρα να ικανοποιήσει το στόχο με διαφορετικό τρόπο. Στην προσπάθειά της αυτή, επαναδρομεί στον αμέσως προηγούμενο υποστόχο που ικανοποίησε, αρέσει (Y, κρασί), με σκοπό να τον ενοποιήσει μ'ένα διαφορετικό τύπο Horn του προογράμματος. Σ'αυτό το σημείο, η Prolog αποδεσμεύει όλες τις μεταβλητές από τις ενημερώσεις τους. Έπειτα, ελέγχει τους προγραμματικούς τύπους της βάσης δεδομένων που βρίσκονται πιο βαθιά από τον τελευταίο προγραμματικό τύπο με τον οποίο ενοποίησε τον υποστόχο, βήμα 6, προγραμματικός τύπος 3, όπου πιο βαθιά σημαίνει για την Prolog ότι ο αντίστοιχος τύπος έχει μεγαλύτερο αύξοντα αριθμό από τον τελευταίο τύπο με τον οποίο έγινε η ενοποίηση. Επειδή σε κάθε υπολογιστικό βήμα, η θέση του υπό εκτέλεση προγραμματικού τύπου της βάσης δεδομένων καταγράφεται, η Prolog έχει την δυνατότητα να εντοπίζει τους πιο βαθείς προγραμματικούς τύπους. Επειδή δεν υπάρχει κανένα γεγονός και κανένας κανόνας, των οποίων η κεφαλή ενοποιείται με τον υποστόχο, η Prolog τερματίζει την έρευνά της.

Στην παραπάνω περιγραφή της εκτέλεσης του προογράμματος, φαίνεται η προσπάθεια της Prolog να θέτει προς ικανοποίηση ένα συγκεκριμένο στόχο σε κάθε υπολογιστικό βήμα. Αυτή η ιδιότητα του συμπερασματικού μηχανισμού της Prolog και ειδικότερα της διαδικασίας επαναδρόμησης δίνει στην Prolog έναν αιτιοκρατικό (deterministic) χαρακτήρα.

Η σειρά με την οποία έχουν εισαχθεί οι προγραμματικοί τύποι στη βάση δεδομένων παίζει βασικό ρόλο στην εκτέλεση και τον έλεγχο ενός Prolog προογράμματος. Η προσπάθεια της Prolog να ικανοποιεί τους (υπο)στόχους του

προγράμματος ερευνώντας σε βάθος τη βάση δεδομένων, υλοποιεί μία από τις βασικές αλγοριθμικές στρατηγικές έρευνας δένδρων (tree searching strategies), την πρώτα-σε-βάθος έρευνα (depth-first search strategy).

## ΣΥΣΤΗΜΑΤΑ - ΕΜΠΕΙΡΟΓΝΩΜΟΝΕΣ

Η **Τεχνητή Νοημοσύνη**, TN,(artificial intelligence), ασχολείται με την ανάπτυξη και τυποποίηση ευφυών μεθόδων και διαδικασιών επίλυσης προβλημάτων, όπου με τον όρο “ευφυείς” μέθοδοι εννοούμε μεθόδους που επιτρέπουν συλλογισμούς και κρίσεις, ανάλογους κατά κάποιον τρόπο μ'αυτούς της ανθρώπινης σκέψης. Η έρευνα στην Τεχνητή Νοημοσύνη στοχεύει στην κατασκευή συμβολικών αναπαραστάσεων του κόσμου και καλύπτει περιοχές όπως η ρομποτική, η κατανόηση φυσικών γλωσσών και τα **Συστήματα-Εμπειρογνώμονες**.

### Συστήματα-Εμπειρογνώμονες και διαχείριση γνώσεων

Τα **Συστήματα-Εμπειρογνώμονες** (expert systems) είναι προγράμματα υπολογιστή ικανά να επιλύουν σύνθετα προβλήματα τα οποία απαιτούν μεγάλες γνώσεις και πείρα για την επίλυσή τους [BS84]. Η σημαντικότερη διαφορά ανάμεσα στα παραδοσιακά προγράμματα και τα Συστήματα-Εμπειρογνώμονες βρίσκεται στο ότι τα παραδοσιακά προγράμματα διαχειρίζονται δεδομένα, ενώ τα Συστήματα-Εμπειρογνώμονες διαχειρίζονται γνώσεις. Αναλυτικά, οι διαφορές μεταξύ των δύο τύπων προγραμμάτων φαίνονται στον παρακάτω πίνακα:

Διαχείριση Δεδομένων	Διαχείριση Γνώσεων
Αναπαράσταση & χρήση δεδομένων	Αναπαράσταση & χρήση γνώσεων
Αλγοριθμικές Διαδικασίες	Ευρετικές Διαδικασίες
Επαναλαμβανόμενες Διαδικασίες	Συμπερασματικές Διαδικασίες

Τα συστήματα -Εμπειρογνώμονες είναι συστήματα ειδικής εφαρμογής, με την έννοια ότι αναφέρονται σε υποπεριοχές ειδικών πεδίων εφαρμογών,

όπως τα ιατρικά διαγνωστικά συστήματα, συστήματα διάγνωσης μηχανικών βλαβών, συστήματα σύνθεσης πολύπλοκων χημικών ενώσεων κ.λ.π.

Τα μέσα με τα οποία τα Συστήματα-Εμπειρογνώμονες επιτυγχάνουν τους στόχους τους στηρίζονται σε σύνολα γεγονότων (facts) και σε σύνολα ευρετικών κανόνων (heuristics) Γεγονότα και κανόνες προέρχονται από ειδικούς επιστήμονες που ασχολούνται με τον εκάστοτε τομέα στον οποίον αναφέρεται το συγκεκριμένο Σύστημα-Εμπειρογνώμονας. Το σύνολο των γεγονότων και ευρετικών κανόνων μπορούμε να πούμε ότι αποτελεί τις γνώσεις του Συστήματος-Εμπειρογνώμονα στο συγκεκριμένο πεδίο, γι'αυτό τα Συστήματα-Εμπειρογνώμονες συνήθως να ονομάζονται και Συστήματα Βασισμένα σε Γνώσεις (Knowledge Based Systems).

Ένα Σύστημα-Εμπειρογνώμονας αποτελείται από τη βάση δεδομένων, το μηχανισμό συμπερασμάτων και το διάμεσο επικοινωνίας χρήστη-συστήματος.

**Η Βάση Γνώσεων** είναι ένα σύνολο γεγονότων και καταστάσεων, το οποίο αποτελεί την περιγραφή του σύμπαντος του συγκεκριμένου πεδίου. Η αναπαράσταση των γνώσεων [Wat 86] σε συστήματα βασισμένα σε κανόνες γίνεται με τη χρήση κανόνων της μορφής

```
IF <προυπόθεση_1> και...και <προϋπόθεση_κ>
THEN <πράξη>
```

**Ο Μηχανισμός Συμπερασμάτων** αποτελείται από ένα σύνολο κανόνων γενικής εφαρμογής, το οποίο χρησιμοποιείται για την κατεύθυνση και τον έλεγχο της διαδικασίας εξαγωγής συμπερασμάτων. Ο μηχανισμός συμπερασμάτων υλοποιεί γενικά μοντέλα που χαρακτηρίζουν και τυποποιούν τις διαδικασίες επεξεργασίας και επίλυσης προβλημάτων, όπως η πρώτα-σε-βάθος έρευνα, η επαναδρομήση και άλλες γενικές συμπερασματικές διαδικασίες που έχουν περιγραφεί σε προηγούμενες ενότητες.

Στις περιπτώσεις που ο μηχανισμός συμπερασμάτων μπορεί να αυτονομηθεί από τις ειδικές γνώσεις που περιέχει ένα συγκεκριμένο Σύστημα-Εμπειρογνώμονας και να λειτουργήσει ανεξάρτητα, μπορεί να χρησιμοποιηθεί για την ανάπτυξη νέων Συστημάτων-Εμπειρογνώμονων.

Τέτοια συστήματα γενικής συμπερασματικής ισχύος ονομάζονται **κελύφη** (shells)

Το **Διάμεσο** είναι υπεύθυνο για την επικοινωνία του χρήστη με το σύστημα. Στις περισσότερες περιπτώσεις χρησιμοποιούνται ειδικά υποσυστήματα ανάλυσης και σύνθεσης φυσικής γλώσσας (parsers) με σκοπό την φυσική και φιλική επικοινωνία μεταξύ χρήστη και συστήματος.

Η απόκτηση και τυποποίηση των γνώσεων που χρειάζεται να ενσωματωθούν σε ένα συγκεκριμένο Σύστημα-Εμπειρογνώμονα, είναι μία από τις βασικότερες και ταυτόχρονα δυσκολότερες φάσεις κατά την ανάπτυξη του συστήματος γι' αυτό και ονομάζεται "ο λαιμός του μπουκαλιού της απόκτησης γνώσεων" (knowledge acquisition bottleneck). Η δυσκολία βρίσκεται στην αδυναμία εύρεσης και ταυτόχρονα τυποποίησης των συγκεκριμένων κανόνων που εκφράζουν τις διαδικασίες που ακολουθούν οι ειδικοί κατά την επίλυση προβλημάτων του πεδίου. Έτσι η δουλειά του μηχανικού γνώσεων, δηλαδή του ανθρώπου που είναι επιφορτισμένος να βρει και να διατυπώσει τις διαδικασίες αυτές, είναι το βασικό στοιχείο για την επιτυχή ανάπτυξη και χρήση ενός Συστήματος-Εμπειρογνώμονα.

Τα τελευταία χρόνια, λόγω της σημασίας των διαδικασιών απόκτησης γνώσεων, έχουν αναπτυχθεί συστήματα αυτόματης εξαγωγής γνώσεων (automatic knowledge acquisition systems). Οι μέθοδοι που χρησιμοποιούνται έχουν σε αρκετές περιπτώσεις μεγάλη επιτυχία και παρουσιάζουν μεγάλο θεωρητικό ενδιαφέρον [RR87, Qui86].

Τα **μέσα** (tools) που χρησιμοποιούνται για την ανάπτυξη Συστημάτων-Εμπειρογνομώνων χωρίζονται σε δύο μεγάλες κατηγορίες [WK83]: **τις υψηλού επιπέδου γλώσσες επεξεργασίας συμβόλων και τα συστήματα-εμπειρογνώμονες γενικής εφαρμογής, τα κελύφη.**

Οι πιο διαδεδομένες γλώσσες επεξεργασίας συμβόλων υψηλού επιπέδου είναι οι LISP [WB81] και Prolog.

Τα κελύφη χρησιμοποιούνται για τη διευκόλυνση και την επιτάχυνση της ανάπτυξης Συστημάτων-Εμπειρογνομώνων ειδικής εφαρμογής. Με τη βοήθεια

των ειδικών φιλικών διάμεσων και κατασκευών που ενσωματώνουν, διευκολύνουν την εισαγωγή αλλά και την αναθεώρηση, αν χρειαστεί, των γεγονότων και των κανόνων ενός Συστήματος-Εμπειρογνώμονα. Ακόμα, ενσωματώνουν έναν αυτόνομο μηχανισμό εξαγωγής συμπερασμάτων, ικανό να συνδιαλλαγεί και να εξάγει συμπεράσματα από διαφορετικές βάσεις γνώσεων, προσφέροντας έτσι τη δυνατότητα μιας γρήγορης ανάπτυξης πρωτότυπων Συστημάτων-Εμπειρογνομώνων.

## Ένα Σύστημα-Εμπειρογνώμονας για νεφρικές παθήσεις

Το Σύστημα-Εμπειρογνώμονας Νεφρά, ΣΕΝ, είναι ένα σύστημα διάγνωσης νεφρικών παθήσεων που υλοποιήθηκε με τη συνεργασία της Ομάδας Λογικής και Λογικού Προγραμματισμού του Μαθηματικού τμήματος στο Πανεπιστήμιο Πατρών και ομάδας ειδικών γιατρών από το Πανεπιστημιακό Νοσοκομείο Πατρών.

Στην παρουσίαση που ακολουθεί, οι προγραμματικοί τύποι και οι διαδικασίες που υλοποιούν το ΣΕΝ έχουν περιοριστεί σ'εκείνους τους προγραμματικούς τύπους και διαδικασίες που περιγράφουν τη δομή και λειτουργία του συστήματος και ταυτόχρονα δείχνουν τη χρήση της Prolog στην κατασκευή Συστημάτων-Εμπειρογνομώνων.

Οι κανόνες που αποτελούν την ειδική Βάση Γνώσεων του ΣΕΝ είναι τυποποιήσεις του τρόπου με τον οποίο ένας γιατρός κάνει διάγνωση με βάση τα συμπτώματα του ασθενούς: αν ο ασθενής είναι σχετικά μεγάλος σε ηλικία, τα ούρα του παρουσιάζουν μεταβολή ποιότητας και αίμα και τα νεφρά του φαίνονται διογκωμένα, τότε ο ασθενής έχει κατά πάσα πιθανότητα υδρονέφρωση. Έτσι, οι κανόνες του ΣΕΝ έχουν την παρακάτω γενική μορφή

IF τα στοιχεία του ασθενή ικανοποιούν συγκεκριμένες απαιτήσεις  
και αναφέρεται(X)

και/ή διαπιστώνεται <σύμπτωμα\_1>



και/ή.....

.....

και/ή διαπιστώνεται <σύμπτωμα\_κ>

και/ή βρίσκεται <εργαστηριακό\_εύρημα\_1>

και/ή βρίσκεται <εργαστηριακό\_εύρημα\_λ>

THEN η διάγνωση είναι <ασθένεια> με πιθανότητα Π

Οι κανόνες αυτοί σχηματίστηκαν με τη βοήθεια και καθοδήγηση των ειδικών γιατρών. Έτσι για την υδρονέφρωση, ο αντίστοιχος κανόνας είναι διάγνωση(υδρονέφρωση, 0.7) :-

ηλικία(H),  $H > 11$ ,

αναφέρεται(μεταβολή\_ποιότητας\_ούρων)

διαπιστώνεται(αιματοουρία)

διαπιστώνεται(αμφοτερόπλευρη\_διόγκωση\_νεφρού).

όπου 0.7 είναι η πιθανότητα να έχει ο συγκεκριμένος ασθενής υδρονέφρωση. Το πεθανοθεωρητικό μέτρο που προσαρτάται σε κάθε διάγνωση είναι ένα εκ των προτέρων ορισμένο μέτρο πιθανολόγησης της διάγνωσης. Έτσι προσφέρεται η διάγνωση της ίδιας ασθένειας μέσα από διαφορετικούς δρόμους και με διαφορετικές πιθανολογήσεις.

Τα βασικά κατηγορήματα που ενσωματώνει το ΣΕΝ είναι:

<b>στοιχεία :</b>	Δέχεται τέσσερα ορίσματα, το φύλο, την ηλικία, το όνομα και το επίθετο του ασθενούς.
<b>διαπιστώνεται :</b>	Δέχεται ένα όρισμα, την ονομασία του συμπτώματος που διαπιστώνεται από τον εξετάζοντα γιατρό.
<b>βρίσκεται :</b>	Δέχεται ένα όρισμα, την ονομασία του αποτελέσματος μιας πιθανής εργαστηριακής

- ανάλυσης.
- αναφέρεται :** Δέχεται ένα όρισμα, ένα από τα “πόνος” “μεταβολή\_ποιότητας\_ούρων” “μεταβολή\_ποσότητας\_ούρων”. “μεταβολή\_συχνότητας\_ουρήσεων”. Το κατηγορημα “αναφέρεται” χρησιμοποιείται για την ομαδοποίηση των συμπτωμάτων.
- Έτσι, ένας κανόνας της παραπάνω γενικής μορφής που δεν έχει στο σώμα του το κατηγορημα “αναφέρεται”, με οποιοδήποτε όρισμα, δεν ενεργοποιείται, με αποτέλεσμα την επιτάχυνση της διαδικασίας εξαγωγής συμπερασμάτων.
- διάγνωση :** Δέχεται δύο ορίσματα, την ονομασία της διάγνωσης και την αντίστοιχη πιθανότητα να έχει ο ασθενής την ασθένεια που δηλώνει η διάγνωση.

Μετά την περιγραφή των βασικών κατηγορημάτων και των ειδικών κανόνων του ΣΕΝ, μπορούμε να προχωρήσουμε στην παρουσίαση των διαδικασιών και των αντίστοιχων προγραμματικών τύπων που υλοποιούν τον συμπερασματικό μηχανισμό του συστήματος.

Στην αρχή πρέπει να ορίσουμε την διαδικασία εισαγωγής των στοιχείων του ασθενή, δηλαδή το κατηγορημα **στοιχεία:**

```
στοιχεία(Φ,Η,Ε,Ο,) : -write(“Φύλο ασθενή (α/γ)”), read(Φ), nl,
                    write(“Ηλικία ασθενή”), read(H), nl,
                    write(“Επίθετο ασθενή”), read(E),nl,
                    write(“Όνομα ασθενή”), read(O),nl,
                    assertz(φύλο(Φ)),
                    assertz(ηλικία(H)),
```

```
assertz(επίθετο(E)),
assertz(όνομα(O)).
```

Στη συνέχεια πρέπει να οριστεί η διαδικασία εισαγωγής των συμπτωμάτων που αναφέρονται από τον ασθενή, δηλαδή το κατηγορημα **αναφέρεται**:

```
αναφέρεται(X) : -write("Αναφέρεται από τον ασθενή “,X,” (v/o) ; “),
read(ANSWER), nl,
ANSWER = “v”
assertz(αναφέρεται(X)).
```

Τώρα θα πρέπει να οριστούν οι διαδικασίες εισαγωγής των συμπτωμάτων που διαπιστώνει ο γιατρός κατά την εξέταση, καθώς και των πιθανών εργαστηριακών αποτελεσμάτων που μπορεί να έχει ο γιατρός στη διάθεσή του.

```
διαπιστώνεται(Σ) : -write(Διαπιστώνεται κατά την εξέταση “,Σ,” (v/o) ;”),
read(ANSWER), nl,
ANSWER = “v”,
assertz(διαπιστώνεται(Σ)).
```

```
βρίσκεται(A) : -write(“Διαπιστώνεται εργαστηριακά”,A,” (v/o);”),
read(ANSWER), nl,
ANSWER = “v”,
assertz(βρίσκεται(A)).
```

Στους παραπάνω ορισμούς, η ύπαρξη του assertz έχει σαν αποτέλεσμα την εισαγωγή της βασικής υλοποίησης της κεφαλής του αντίστοιχου κανόνα, όταν αυτή ικανοποιηθεί. Έτσι, ο διάλογος μεταξύ συστήματος και χρήστη κατά τη συμπερασματική διαδικασία καταγράφεται και αποθηκεύεται για οποιαδήποτε περαιτέρω επεξεργασία.

Τέλος ορίζουμε το γενικό διαγνωστικό κατηγορημα:

```
προχώρησε_σε_διάγνωση : -!, στοιχεία(Φ,Η,Ε,Ο),
διάγνωση(Δ,Π),
assertz(διάγνωση(Δ,Π)).
```

Αν το παραπάνω κατηγορημα τεθεί ως ερώτημα και υποθέσουμε ότι η βάση δεδομένων ενσωματώνει διαγνωστικούς κανόνες, η Prolog θα προχωρήσει στην εξαγωγή συμπερασμάτων. Ουσιαστικά θα υπάρξει ένας διάλογος μεταξύ του χρήστη και της Prolog με αποτέλεσμα τις διαγνώσεις που μπορούν να επιβεβαιωθούν, ανάλογα με τις απαντήσεις του χρήστη.

Ας υποθέσουμε π.χ., ότι η βάση δεδομένων περιέχει τους ακόλουθους διαγνωστικούς κανόνες:

διάγνωση(οξεία\_πυελονεφρίτιδα, 0.7) :-

αναφέρεται(πόνος),  
 διαπιστώνεται(υψηλός\_πυρετός\_με\_ρίγος),  
 διαπιστώνεται(πυουρία).

διάγνωση(οξεία\_πυελονεφρίτιδα, 0.8) :-

ηλικία(H), H<9,  
 διάγνωση(οξεία\_πυελονεφρίτιδα, 0.7),  
 βρίσκεται(θετική\_ουροκαλλιέργια).

διάγνωση(οξεία\_πυελονεφρίτιδα, 0.9) :-

αναφέρεται(μεταβολή\_συχρότητας\_ουρήσεων

)  
 αναφέρεται(πόνος)  
 διάγνωση(οξεία\_πυελονεφρίτιδα, 0.8),  
 διαπιστώνεται(πόνος\_στα\_νεφρά),  
 διαπιστώνεται(συχνουρία),  
 διαπιστώνεται(δυσουρία).

Θέτοντας τώρα το ερώτημα “? προχώρησε\_σε\_διάγνωση. “γίνεται ο ακόλουθος διάλογος, όπου οι απαντήσεις του χρήστη είναι υπογραμμισμένες:

? Φύλο ασθενή (α/γ). α

? Ηλικία ασθενή. 11

? Επίθετο ασθενή. ΕΠΙΘΕΤΟ

? Όνομα ασθενή. ΟΝΟΜΑ

----- η Prolog εισάγει τους προγραμματικούς  
τύπους

φύλο(**a**).

ηλικία(**11**).

επίθετο(**ΕΠΙΘΕΤΟ**).

όνομα(**ΟΝΟΜΑ**).

-----

? Αναφέρεται από τον ασθενή πόνος (v/o). γ

----- η Prolog εισάγει τον τύπο  
αναφέρεται(πόνος).

-----

? Διαπιστώνεται κατά την εξέταση υψηλός\_πυρετός\_με\_ρίγος (v/o). γ

----- η Prolog εισάγει τον τύπο  
διαπιστώνεται(υψηλός\_πυρετός\_με\_ρίγος).

-----

? Διαπιστώνεται κατά την εξέταση πνιγμός (v/o). γ

----- η Prolog εισάγει τους τύπους  
διαπιστώνεται(πνιγμός).

διάγνωση(οξεία\_πνευμονοπνευμονίτιδα, 0.7).

και προσπαθεί να ικανοποιήσει το

“διάγνωση” με διαφορετικό τρόπο.

Επειδή στους όρους του δεύτερου τύπου

για το “διάγνωση”, αναφέρεται το “ηλικία

(H)”, το οποίο ικανοποιείται για  $H=11$ ,

ο όρος  $H<9$  δεν ικανοποιείται και έτσι η

Prolog προχωρεί στην εξέταση του τρίτου

τύπου για το κατηγορημα διάγνωση.

-----

? Αναφέρεται από τον ασθενή μεταβολή\_συχνότητας\_ουρήσεων (v/o). ο

-----η Prolog εγκαταλείπει και αυτόν τον κανόνα και τερματίζει τη διαδικασία ικανοποίησης του αρχικού ερωτήματος και απαντάει λόγω της τομής

---

$\Delta$ =οξεία\_πυελονεφρίτιδα

$\Pi$ =0.7

## **ΜΕΡΟΣ Β**

## ΕΦΑΡΜΟΓΗ ΕΞΥΠΝΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Οι εφαρμογές που θα αναφέρουμε ανήκουν στην περιοχή της expert systems.

### Τι είναι ένα expert system?

Όλοι γνωρίζουμε ότι ακόμα και σήμερα οι computer designers (σχεδιαστές υπολογιστών) προσπαθούν να κάνουν τους υπολογιστές ευκολότερους στην χρήση ή καλύτερα προσπαθούν να γεφυρώσουν το κενό ανάμεσα στον άνθρωπο και στην μηχανή. Αλλά ακόμα και σήμερα ένας computer είναι μόνο μια μηχανή.

Δεν μπορούμε (ακόμα) να επικοινωνήσουμε μαζί του στα ελληνικά ή στα Αγγλικά. Δεν μπορούμε να απαιτήσουμε να μας λύσει κάποιο πρόβλημά μας παρά μόνο αν ήδη τον έχουμε εφοδιάσει (προγραμματίσει) με μια κατάλληλη μέθοδο λύσης.

Δεν μπορούμε να περιμένουμε να μάθει από τα λάθη του και να χρησιμοποιήσει προηγούμενη εμπειρία για την λύση προβλημάτων. Ή μήπως μπορούμε?

Τα τελευταία χρόνια αναπτύχθηκε μια περιοχή έρευνας της T.N. που ονομάστηκε expert systems. Στην πραγματικότητα τα expert systems είναι μια από τις λίγες περιοχές της τεχνητής νοημοσύνης που βγήκε από τα ερευνητικά εργαστήρια και εφαρμόστηκε και αναπτύχθηκε στον πραγματικό κόσμο (εφαρμογή σε Βιομηχανίες, Νοσοκομεία, κ.λ.π.)

### Ορισμός

**Ένα expert system είναι ένα computer system το οποίο ενεργεί, όπως ένας expert. Ενσωματώνει γνώση (specific problem domain) και κατέχει την ικανότητα να εφαρμόζει αυτήν την γνώση.**



## ΚΑΤΑΛΟΓΟΙ ΕΦΑΡΜΟΓΩΝ

### 1. ΓΕΩΡΓΙΑ.

Το PLANT/cd προλέγει τις καταστροφές που γίνονται στις φυτείες καλαμποκιών και οφείλονται στην black cutworm. Η γνώση στην "τράπεζα γνώσης" του συστήματος αναπαρίσταται με rules στους οποίους χρησιμοποιείται σαν μέθοδος εξαγωγής συμπερασμάτων backward chaining control μηχανισμός. Το σύστημα παίζει πόλο "συμβουλάτορα" και έχει αναπτυχθεί στο University of Illinois. Το PLANT/ds παρέχει ένα "συμβούλιο" που αναφέρεται στις διαγνώσεις ασθενειών που μπορεί να πλήξουν φυτείες από soybeans (είδος οσπρίων). Η γνώση στο σύστημα βρίσκεται υπό μορφή (rules) κανόνων με δύο τύπους.

**1. Κανόνες που αναπαριστούν την διαγνωστική εμπειρία του συστήματος.**

**2. Κανόνες που αποκτιούνται (obtained) μέσω του AQ11 που είναι ένα inductive inference program.**

Το PLANT/ds είναι ένα μέρος του ADVISE και δημιουργήθηκε στο Πανεπιστήμιο του Illinois.

POMME Βοηθά τους αγρότες να διευθύνουν καλλιέργειες μήλων προμηθεύοντάς τους με συμβουλή στο πως να βελτιώνουν την συγκομιδή των μήλων. Οι ειδικοί που έδωσαν την "γνώση" στο σύστημα αυτό το σύστημα Βοτανολόγοι και Εντομολόγοι. Η αναπαράσταση της γνώσης σ αυτό το σύστημα γίνεται με rules και frames. Είναι φτιαγμένο σε P R O L O G και

τρέχει σε V A X 11/780. Τα P O M M E δημιουργήθηκαν στο Virginia Polytechnic Institute and State University.

## 2. ΧΗΜΕΙΑ.

Το CONGEN βοηθά τους κατασκευαστές χημικών ενώσεων να προσδιορίζουν ένα σύνολο από δυνατές structures για μια άγνωστη σύνθεση. Ο χημικός εφοδιάζει το CONGEN με spectroscopic και chemical data καθώς και με ένα σύνολο από required και forbidden constraints (περιορισμοί) πάνω στις δυνατές ενδοσυνδέσεις μεταξύ των ατόμων της σύνθεσης. Το CONGEN βρίσκει όλους εκείνους τους δυνατούς τρόπους που μπορεί να συναρμολογηθούν τα άτομα μέσα στις molecular structures οι οποίοι ικανοποιούν τους "επιβληθέντες" περιορισμούς. Στην συνέχεια το CONGEN δίνει αναλυτικά στον χημικό τις εικόνες των υποψήφιων κατασκευών (candidate structures). Το σύστημα παράγει υποψήφιες (candidate) κατασκευές χρησιμοποιώντας μία ποικιλία γραφο-θεωρητικών αλγορίθμων. Το CONGEN εκτελείται σε INTERSLIP. Αναπτύχθηκε στο Πανεπιστήμιο του Στάνφορντ σαν μέρος του DENTRAL και χρησιμεύει σαν γεννήτρια υπόθεσης της Ευριστικής του DENTRAL. Το CONGEN έφθασε το στάδιο μιας πρωτότυπης έρευνας.

Το CRYSALIS συμπεραίνει την 3-διαστατική κατασκευή μιας πρωτεΐνης από ένα χάρτη πυκνότητας ηλεκτρονίων (ΧΠΗ). Το σύστημα ερμηνεύει τα δεδομένα διάθλασης ακτίνας X που προσδιορίζονται από θέση και ένταση διαθλαθέντων κυμάτων για να συμπεραίνει αυτή την ατομική κατασκευή. Το σύστημα χρησιμοποιεί γνώση σχετικά με σύνθεση πρωτεϊνών και κρυσταλλογραφία ακτίνων X και ευριστικά για ανάλυση των ΧΠΗ για να παράγει και να δοκιμάζει υποθέσεις σχετικές με ευλογοφανή κατασκευή

πρωτεΐνης. Το CRYVALIS χρησιμοποιεί blackboard αρχιτεκτονική, που περιέχει ανεξάρτητες πηγές γνώσεων που κτίζουν και δοκιμάζουν μία κατασκευή υπόθεσης πολλών επιπέδων. Το σύστημα εκτελείται με LISP. Αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ και έφθασε στο στάδιο ερευνής πρωτοτύπων.

Το C-13 βοηθά οργανικούς χημικούς στο να καθορίζουν την κατασκευή καινούργιων συνθέσεων. Το C-13 βοηθά τον χημικό να αναλύσει τον άνθρακα-13 πυρηνική μαγνητική φασματική απήχηση, χρησιμοποιώντας ένα search για να αποφασίσει την οργάνωση ατόμων και δεσμών πολύπλοκων οργανικών μορίων. Η βάση της γνώσης του συστήματος περιέχει κανόνες που έχουν σχέση με φασματικά χαρακτηριστικά (resonance) απήχησης που παράγονται αυτόματα από δεδομένα για γνωστές κατασκευές. Το C-13 αναπτύχθηκε σαν μέρος του σχεδίου DENDRAL και ακολουθεί το παράδειγμα του σχεδίου παραγωγής και δοκιμής του DENDRAL. Το σύστημα υλοποιήθηκε σε INTERSLIP. Αναπτύχθηκε στο Πανεπιστήμιο του Στάνφορντ και έφθασε στο στάδιο πρωτοτύπου έρευνας.

Το DENDRAL συμπεραίνει τη μοριακή κατασκευή άγνωστων συστατικών από φασματική μάζα και πυρηνικά μαγνητικά δεδομένα. Το σύστημα χρησιμοποιεί έναν ειδικό αλγόριθμο αναπτυγμένο από τον J.Lederberg που απαριθμεί συστηματικά όλες τις πιθανές μοριακές κατασκευές και χρησιμοποιεί χημική εμπειρογνωμοσύνη για να κλαδεύσει αυτόν τον κατάλογο πιθανοτήτων δε ένα κατορθωτό μέγεθος. Η γνώση στο DENDRAL αναπαρίσταται: (1) με την μορφή procedural code για την γεννήτρια μοριακής κατασκευής και (2) υπό την μορφή rules για την data-driven συνιστώσα και την αξιολόγηση. Το σύστημα υλοποιήθηκε με INTERSLIP και αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ. Έχει φθάσει το στάδιο παραγωγής πρωτοτύπου.

Το GA1 αναλύει την DNA κατασκευή από δεδομένα περιορισμού διαχώρισης ενζύμων. Το σύστημα δέχεται δεδομένα επιτομής ενζύμων (τοπολογία, ανεκτικότητα κ.λ.π.). Παράγει όλες τις υποψήφιες κατασκευές και τελικά αποφασίζει (καθορίζει) ευλογοφανείς κατασκευές DNA. Η εμπειρογνωμοσύνη που περιείχετο στο GA1 είναι ένα βιβλίο κειμένων, μοντέλο του μηχανισμού που περιλάμβανε στην ανάλυση χώνευσης ενζύμων κατασκευών του DNA. Το GA1 ακολουθεί για παραγωγή και τεστ παράδειγμα που χρησιμοποιείται στο DENDRTAL. Οι υποθέσεις προτείνονται από μια γεννήτρια βασισμένη σε πορεία για απαρίθμηση όλων των πιθανών λύσεων. Το σύστημα παρέχει μια αντιλογία προσαρμόσιμου χρήστη επιπέδου ανεκτικότητας για αποζημιώσει μικρές ποσότητες εσφαλμένων δεδομένων. Το GA1 γράφεται σε INTERLISP. Αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ και έφθασε το στάδιο μιας έρευνας πρωτότυπου.

Το META-DENDRAL βοηθά τους χημικούς να υπολογίσουν την εξάρτηση της mass spectrometric fragmentation στα υποικοδομητικά χαρακτηριστικά. Αυτό γίνεται με την ανακάλυψη κανόνων διατεμάχισης (fragmentation) για δοσμένες τάξεις μορίων. Το σύστημα παράγει αυτούς τους κανόνες από υποδείξεις (παραδείγματα) εκπαίδευσης που αποτελούνται από σετ μορίων με γνωστές 3-διαστατικές κατασκευές και μαζικά φάσματα. Το META-DENDRAL πρώτα παράγει ένα σετ υψηλά-ειδικών κανόνων που υπολογίζουν μοναδική πορεία τεμάχισης σε ένα ειδικό μόριο. Κατόπιν χρησιμοποιεί τα παραδείγματα εκπαίδευσης για να γενικεύσει αυτούς τους κανόνες. Τελικά, το σύστημα επανεξετάζει τους κανόνες για να μετακινήσουν πλεονάζοντες ή λάθος κανόνες. Το META-DENDRAL εκτελείται σε INTERLISP. Αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ και έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το MOLGEN βοηθά τον γυναικολόγο να σχεδιάσει gene-cloning πειράματα σε μοριακή γενεσιολογία. Αυτά τα πειράματα εμπλέκουν (περιλαμβάνουν) σύνδεση γένους που αποκρυπτογραφεί ένα επιθυμητό προϊόν πρωτεΐνης σε βακτήρια έτσι ώστε τα βακτήρια να το κατασκευάσουν. Το σύστημα χρησιμοποιεί γνώσεις για την (γενετική) και το αποτέλεσμα του είναι να δημιουργήσει ένα αφηρημένο σχέδιο που κατόπιν θα το "εκλεπτύνει" σε σετ ειδικών εργαστηριακών βημάτων. Το MOLGEN χρησιμοποιεί αναπαράσταση γνώσης (1) βασισμένη στην μέθοδο του (game) και (2) ένα control scheme. Υλοποιήθηκε σε LISP και UNITS και αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ. Το MOLGEN είναι πρωταρχικά ένα όχημα για να δοκιμάζει προσεγγίσεις συζητώντας, για σχέδιο μάλλον παρά ένα έμπειρο σύστημα εγχείρησης για μοριακή γενετική.

Το OCSS βοηθά τους χημικούς στο να συνθέτουν πολύπλοκα οργανικά μόρια. Το σύστημα αναλύει στόχους μοριακών (επινοούμενα από τον χημικό με αναγνώριση λειτουργικών ομάδων/γκρουπ) αλυσίδων, δαχτυλιδιών και πλεονασμού ή συμμετρίας στο μοριακό σκελετό, εφαρμόζοντας χημικές τροποποιήσεις σ' αυτό και αξιολογώντας την τελική κατασκευή για διόρθωση, μοναδικότητα και απλότητα. Το σύστημα εκτελείται σε DECPDP-1 computer στο Πανεπιστήμιο Χάρβαρντ. Έφθασε το στάδιο έρευνας πρωτότυπου.

Το SECS βοηθά τους χημικούς στο να συνθέσουν περίπλοκα οργανικά μόρια. Ο χημικός παρουσιάζει κατασκευή μοριακού στόχου και το σύστημα παράγει σχέδιο για να δημιουργήσει μοριακό στόχο από βασικό κτίσιμο μορίων. Το σχέδιο είναι σειρά χημικών αντιδράσεων που εφαρμόζονται σε λειτουργικά γκρουπ ατόμων. Το σύστημα με τη βοήθεια του χημικού, εργάζεται "προς τα πίσω" από το στόχο στην οικοδόμηση. Το SECS εκτελείται σε FORTRAN. Αναπτύχθηκε στο Πανεπιστήμιο της Καλιφόρνια, Σάντα Κρουζ και έφθασε το στάδιο ερευνής πρωτοτύπου.

Το SEQ βοηθά μοριακούς βιολόγους που εκτελούν διάφορους τύπους πυρηνικής σειράς ανάλυσης. Το σύστημα μπορεί να αποθηκεύσει και να αναλύσει σειρές nucleic acid (πυρηνικά οξέος). Τον SEQ οι ρουτίνες "ψαξίματος" (searching) μπορούν να χρησιμοποιούν ένα σετ default παραμέτρων. Για παράδειγμα ο βιολόγος μπορεί να ποικίλει τα βάρη των penalties και να εκτιμήσει το χάσμα των αποτελεσμάτων, κατά την διάρκεια της ευθυγραμμίας του Needleman-Wunch. Το SEQ εκτελείται σε LISP. Αναπτύχθηκε σαν μέρος του σχεδίου MOLGEN στο Πανεπιστήμιο Στανφορντ και μετά ολοκληρώθηκε από την Ιντελικορπ. Έφθασε το στάδιο εμπορικού συστήματος.

Το SPEX βοηθά τους επιστήμονες στο να σχεδιάσουν πολύπλοκα εργαστηριακά πειράματα. Ο επιστήμονας περιγράφει τα αντικείμενα που θα χρησιμοποιηθούν και το σύστημα βοηθά την ανάπτυξη ενός βασικού σχεδίου για να επιτύχει πειραματικό αποτέλεσμα. Μετά, το σύστημα διωλίζει κάθε abstract step στο σχέδιο, κάνοντας τα πιο ειδικά, ενώνοντας αυτά σε τεχνικές και αντικείμενα που είναι αποθηκευμένα στο knowledge base του συστήματος. Αν και το σύστημα έχει δοκιμασθεί αποκλειστικά στην μοριακή βιολογία, δεν περιέχει built-in μοριακής βιολογίας μηχανισμούς, έτσι θα μπορούσε να εφαρμοσθεί σε άλλες προβληματικές περιοχές. Το SPEX εκτελείται σε UNITS, μια frame-based γλώσσα για representation. Το σύστημα αναπτύχθηκε στο Στάνφορντ και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το SYNCHEM συνθέτει πολύπλοκα οργανικά μόρια, χωρίς να απαιτεί αλληλεπίδραση πρωτότυπου. Το SYNCHEM χρησιμοποιεί γνώσεις που αφορούν χημικές αντιδράσεις και έτσι δημιουργεί ένα σχέδιο για ανάπτυξη μοριακού στόχου βασιζόμενα σε ένα σετ που δίνει αρχικά μόρια. Το σύστημα δουλεύει (backward) προς το πίσω, αρχίζοντας με μοριακό στόχο και προσπαθεί να αποφασίσει ποιες αντιδράσεις θα μπορούσαν να το παραγάγουν και τι υλικά/μόρια, θα απαιτούνται. Αυτό συνεχίζεται μέχρι να

βρεθεί μια πορεία σύνθεσης από το στόχο στα αρχικά υλικά. Το σύστημα εκτελείται σε PL/1. Είναι ο προκάτοχος του SYNCHEM 2 και αναπτύχθηκε στο STATE Πανεπιστήμιο της Νέας Υόρκης στο Στονι Μπρουκ.

Το SYNCHEM 2 συνθέτει πολύπλοκα οργανικά μόρια χωρίς βοήθεια ή οδηγία από έναν χημικό. Προσπαθεί να ανακαλύψει μια ευλογοφανή σειρά οργανικών συνθετικών αντιδράσεων που θα γίνουν σετ διαθέσιμων αρχικών υλών μέσα σ' επιθυμητό μοριακό στόχο. Το SYNCHEM 2 χρησιμοποιεί γνώσεις για χημικές αντιδράσεις για να παράγει σχέδιο για δημιουργία μοριακού στόχου από τη βασική οικοδομή των μορίων. Το σύστημα προσπαθεί να βρει για optimal πορεία σύνθεσης από τα αρχικά υλικά στην σύνθεση στόχων εφαρμόζοντας heuristics που περιορίζουν την έρευνα σε μονοπάτια που ικανοποιούν the problem constraints. Αυτοί οι περιορισμοί (constraints) μπορεί να περιλαμβάνουν πληροφορίες για συνθήκες τοξικής αντίδρασης καθώς και την ποιότητα του επιθυμητού προϊόντος. Το σύστημα εκτελείται σε PL/1 Είναι ο διάδοχος του SYNCHEM και αναπτύχθηκε στο Πανεπιστήμιο State University of New York στο Στονι Μπρουκ. Έχει φθάσει το στάδιο του πρωτότυπου.

Το TQMSTUNE fine-tunes μια τριπλή τετραπλή μάζα σπεκτομέτρου (TQMS) ερμηνεύοντας δεδομένα (signal date) από την TQMS. Το σύστημα χρησιμοποιεί γνώσεις σχετικά με το πως ποικίλει ο έλεγχος των οργάνων του TQMS επηρεάζει την ευαισθησία και τις φασματικές διαμορφώσεις. Η γνώση αντιπροσωπεύεται, χρησιμοποιώντας τα frame-based χαρακτηριστικά του KEE, την implementation language TQMSTUNE. Το σύστημα ανεπτύχθη από την Ιντελιορπ και έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

### 3. COMPUTER SYSTEMS.

Το CRIB βοηθά τους μηχανικούς κομπιούτερ και διατηρητές συστήματος να προσδιορίζουν τα σφάλματα στο software ή hardware. Ο μηχανικός δίνει στο σύστημα μια περιγραφή των δικών του παρατηρήσεων σε απλά Αγγλικά όπως όροι CRIB που ταιριάζει αυτό έναντι μιας data base από γνωστά σφάλματα. Με διαδοχικό ταίριασμα ολοένα και μεγαλύτερων γκρουπ (ομάδων) συμπτωμάτων με την περιγραφή, το CRIB φθάνει σε ένα subunit που είναι είτε δυνάμενο να επιδιορθωθεί ή να αντικατασταθεί. Εάν φτάσουμε σε subunit το λάθος δεν "θεραπεύεται", το σύστημα χάνει backtrack (αυτόματα) προς το τελικής απόφασης σημείο και προσπαθεί να βρει άλλο ταίριασμα (match). Το CRIB περιέχει hardware και software διαγνωστική εμπειρία υπό την μορφή "action symptom pairs" που ονομάζονται symptom patterns. Σ' αυτά η (δράση) (actim) σχεδιάζεται για να εξαλείψει το σύμπτωμα από την μηχανή. Το CRIB σχεδιάζει την μηχανή κάτω από την διάγνωση απλής ιεραρχίας των enbunits σ' ένα σημαντικό δίκτυο. Το σύστημα γράφεται σε CORALL 66. Αναπτύχθη σαν μια συνδυασμένη προσπάθεια από την International Computers Limited (ICL), από το Research and Advanced Development Centre (RADC) καθώς από το Πανεπιστήμιο Brunel. Το CRIB έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το DART βοηθά στην διάγνωση σφαλμάτων στα συστήματα hardware τα computer χρησιμοποιώντας πληροφορίες για το design(σχέδιο) της συσκευής που γίνεται η διάγνωση. Το σύστημα δουλεύει απευθείας από πληροφορίες για την intended structure και την αναμενόμενη συμπεριφορά της συσκευής και προσπαθεί να βρει design flaws στις νεοδημιουργηθείσες συσκευές. Το σύστημα έχει εφαρμοστεί σε απλά computer circuits και στην πορεία της τηλεπορείας του IBM 4331. Το DAR χρησιμοποιεί μια ανεξάρτητου συσκευής πορεία συμπεράσματος που είναι παρόμοια με ένα τύπο απόδειξης ανάλυσεως θεωρήματος, όπου το σύστημα προσπαθεί να



παράγει μια απόδειξη που σχετίζεται με την αίτια της κακής λειτουργίας της συσκευής. Το σύστημα εκτελείται με MRS και αναπτύχθηκε στο Πανεπιστήμιο Στάνφορντ. Έφθασε το στάδιο έρευνας πρωτοτύπου.

Το IDT βοηθά έναν τεχνικό να τοποθετήσει τα αντικαταστήσιμα κομμάτια του field που θα έπρεπε να αντικατασταθούν για να καθορίζουν σφάλματα στη PDP 11\03 κομπιούτερς. Το σύστημα χρησιμοποιεί γνώσεις για τι unit υπό δοκιμή. Τέτοιες είναι οι λειτουργίες των συνισταμένων του και η σχέση μεταξύ τους καθώς και ο τρόπος που συλλέγουν και εκτελούν διαγνωστικά τεστ και ερμηνεύουν τα αποτελέσματα. Το σύστημα είναι rule-based χρησιμοποιώντας forward chaining και εκτελείται σε FRANZ LISP σε OPS5.

Το IDT ανεπτύχθη από την Digital Equipment Corporation (DEC) και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το ISA χρονοδρομολογεί εντολές πελατών σε σύστημα κομπιούτερ, έναντι στα τρέχοντα και σχεδιασμένα υλικά κατανομών. Το ISA παίρνει τις εντολές πελατών, συμπεριλαμβανομένων αλλαγές και ακυρώσεις και παράγει ένα schedule date για κάθε εντολή. Επιδεικνύει επιπρόσθετες πληροφορίες για μια προβληματική εντολή συμπεριλαμβανομένων εκτός του προταθέντος προγράμματος και εναλλακτικές προτάσεις για να προγραμματίσουν την εντολή(order). Το ISA περιέχει σημαντική εμπειρία προγραμμάτων, όπως γνώσεις για πιθανές ακυρώσεις εντολών, υλική διαθεσιμότητα και στρατηγικές για relaxing scheduling constraints. Το ISA είναι forward chaining και rule-based σύστημα που εκτελείται σε OPS5.

Αναπτύχθηκε από την Digital Equipment Corporation και χρησιμοποιείται στην DEC για την κατασκευή φυτών. Έφθασε το στάδιο ενός εμπορικού συστήματος.

Το MIXER βοηθά τους προγραμματιστές να γράφουν μικροπρογράμματα για το Texas Instruments TI990 VLSI chip. Δίνοντας μια περιγραφή μικροπρογράμματος, το σύστημα παράγει optimized οριζόντιους μικροκώδικες για το TI990. Το MIXER περιέχει γνώσεις για τι μικροπρόγραμμα στο TI990 που παίρνονται από manuals και από μια ανάλυση μικροκώδικα στο TI990 control ROM. Αυτός συμπεριλαμβάνει γνώσεις για το πως να γράψει σε χάρτες περιγραφές σε σετ ενδιάμεσων λειτουργιών πως να τοποθετήσει κατάλληλους καταλόγους σε μεταβλητές και πως να επεκτείνει ενδιάμεσες λειτουργίες σε σετ μικρολειτουργιών. Το MIXER χρησιμοποιεί αυτές τις γνώσεις για να αποφασίσει ποιες μικρολειτουργίες είναι καλύτερες για την εκτέλεση του μικροπρογράμματος. Το σύστημα αναπαριστά την γνώση σαν κανόνες και δεδομένα με συμπερασματική ελεγχθείσα ενοποίηση και δυναμικό backtracking. Το MIXER εκτελείται με PROLOG. Αναπτύχθηκε στο Πανεπιστήμιο του Τόκυο και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το R1-SOAR εκτελεί αναλογία του έργου σχηματισμού του συστήματος του κομπιούτερ. Χρησιμοποιεί κάποια γνώση που ενυπάρχει στο XCON επίσης καλείται R1 που συσχετίζεται με τον σχηματισμό VAX 11\780 συστήματα κομπιούτερ. Ο σχηματισμός συνεπάγεται το να κοιτάς εντολή πελάτη και να αποφασίζεις τι συστηματικά (στοιχεία) πρέπει να προστεθούν, ώστε να παραχθεί ένα complete operational system subject. Το R1-SOAR εργάζεται εκτελώντας search έρευνα διαμέσου ενός problem space από πιθανές λύσεις με σκοπό να προσδιορίσει μια αποδεκτή. Το r1-soar αναπτύχθηκε μέσα στην αρχιτεκτονική λύση γενικού προβλήματος SOAR με το στόχο της καλύτερης κατανόησης στο πως να ξεπεράσουν τους περιορισμούς και των δυο ειδικών συστημάτων και των γενικών λυτών προβλήματος. Αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon σαν επίδειξη πρωτοτύπου.

Το TIMM\TUNER βοηθά στο να ρυθμίζει το VAX\VMS συστήματα του κομπιούτερ για να μειώσει προβλήματα που μεγαλώνουν με την διαρκή αλλαγή στο computer environment. Αλληλεπιδρά με τον system manager κάνοντας του σειρά από ερωτήσεις που οδηγούν σε συγκεκριμένες πράξεις, όπως η προσάρμοση συστήματος παραμέτρων ή user authorization values, ξαναδιανέμοντας ή ελαττώνοντας την απαίτηση του χρήστη, αλλάζοντας το σχέδιο του software ή αγοράζοντας καινούργιο hardware(σκεύος). Το σύστημα χρησιμοποιεί ένα rule-base σχήμα που δημιουργείται μέσα στο TIMM ένα εμπορικό σύστημα για αυτόματη απόκτηση γνώσης. Το TIMM\TUNET αναπτύχθηκε από την General Research Corporation και έφθασε το στάδιο ενός εμπορικού συστήματος.

Το XCON configures VAX 11\780 συστήματα κομπιούτερ. Από μια εντολή πελάτη, αυτό αποφασίζει τι στοιχεία πρέπει να προστεθούν για να παράγουν ένα complete operational σύστημα και αποφασίζει τις spatial σχέσεις ανάμεσα σε όλες τις συνιστώσες. Το XCON παράγει ένα σετ διαγραμμάτων δεικνύοντας αυτές τις spatial σχέσεις σε τεχνικούς που κατόπιν συναρμολογείται το VAX system. Το XCON χειρίζεται το έργο του configuration εφαρμόζοντας γνώσεις περιορισμών σε σχέσεις συνιστωσών σε procedures for configuring computers. Το σύστημα είναι μη αλληλοεπιδραστικό, είναι rule-base και χρησιμοποιεί forward chaining control scheme. Το XCON εκτελείται σε OPS5 και αναπτύχθηκε διαμέσου συνεργασίας ερευνητών του Πανεπιστημίου Carnegie-Mellon και της (DEC) εις το Hudson της Μασσαχουσέτης. Αυτό το εμπορικό ειδικό σύστημα, σχηματίζει VAX κομπιούτερς σε μια καθημερινή βάση για την DEC και είναι το μεγαλύτερο και πιο ώριμο rule-based expert σύστημα σε λειτουργία.

Το XSEL βοηθά έναν πωλητή να συλλέγει στοιχεία για ένα VAX 11\780 σύστημα κομπιούτερ και βοηθά στο να σχεδιάζει ένα σχέδιο πατώματος γι'αυτούς. Το XSEL συλλέγει ένα Central Processing Unit, πρωταρχική μνήμη,

software, και περιφερειακές συσκευές π.χ. terminals και disk-drives και κατόπιν τα περνάει στο XCON για να επεκταθούν και να "σχηματισθούν" (configured). Το XSEL περιέχει κύρια γνώση για τις σχέσεις ανάμεσα στα συστατικά και τις ποικιλίες εφαρμογές που ένας πελάτης μπορεί να έχει και γνώσεις για το πως να οδηγήσει έναν χρήστη διαμέσου μιας selection process. Το σύστημα είναι αλληλεπιδραστικό είναι rule-based και χρησιμοποιεί forward chaining control scheme. Το XSEL εκτελείται σε OPS5. Αναπτύχθηκε διαμέσου μιας συνεργασίας ανάμεσα σε ερευνητές του Carnegie-Mellon Πανεπιστημίου και της DEC στο Hudson της Μασσαχουσέτης και έφθασε το στάδιο ενός πεδίου πρωτοτύπου.

Το Yes\MVS βοηθά τους χειριστές κομπιούτερ να (εκπέμπουν) προειδοποιούν και να ελέγχουν το MVS operation system. Είναι το πλατιά χρησιμοποιούμενο σύστημα λειτουργίας σε large mainframe IBM computers.

Το Yes\MVS απευθύνεται στις 6 μεγαλύτερες κατηγορίες έργων (καθηκόντων) 1) διατηρώντας επαρκή JES (Job Entry System) queue space, 2) χειρίζοντάς το και στέλνοντας είτε διαταγές στο λειτουργικό σύστημα ή συστάσεις στο console operator.

Το Yes\MVS είναι ένα σύστημα rule-based με forward chaining control scheme. Γράφτηκε σε μια εκτεταμένη version της OPS5 από ένα Expert system group στο IBM Watson Research Center in Yorktown Heights, New York. Οι δημιουργοί του σχεδιάζουν να χρησιμοποιήσουν το Yes\MVS σύστημα σε μια κανονική εμπορική βάση.

## 4. ΗΛΕΚΤΡΟΝΙΚΑ

Το ACE αναγνωρίζει φασαρία στα τηλεφωνικά δίκτυα και συστήνει κατάλληλη επιδιόρθωση και επανορθωτική διατήρηση. Το σύστημα λειτουργεί χωρίς ανθρώπινη παρέμβαση, αναλύονται (αναφορές) περιγραφές διατήρησης που παράγονται σε καθημερινή βάση από το CRAS, μια διευθέτηση επισκευής καλωδίων σε πρόγραμμα κομπιούτερ. Το ACE τοποθετεί τα εσφαλμένα τηλεφωνικά καλώδια, αποφασίζει εάν χρειάζονται προληπτική διατήρηση και συλλέγει τον τύπο της πιο πιθανής διατήρησης για να έχει σωστά αποτελέσματα.

Το ACE κατόπιν αποθηκεύει τις συστάσεις σε μια special data base που οι χρήστες μπορούν access. Το σύστημα αποφασίζει εφαρμόζοντας γνώσεις για τα κέντρα καλωδίων, τις αναφορές (περιγραφές) διατήρησης του CRAS και στρατηγικές ανάλυσης του δικτύου. Είναι μια rule-based σύστημα και έχει forward chaining control scheme. Το ACE εκτελείται σε OPS5 και FRANZ LISP και τρέχει σε AT&T 3B-2 μικροκομπιούτερς τοποθετημένα στα γραφεία αναλυτών καλωδίων. Αναπτύχθηκε στα Bell Laboratories at Whippany του Νιου Τζέρσευ. Το ACE έχει δοκιμασθεί και φθάσει το στάδιο του εμπορικού συστήματος.

Το BDS helps locate faulty modules σε ένα μεγάλο δίκτυο σημάτων, μια ηλεκτροσυσκευή που ονομάζεται baseband distribution subsystem. Το σύστημα χρησιμοποιεί δοκιμαστικό εφοδιασμό ανάγνωσης για να απομονώσει εσφαλμένους τυπωμένους πίνακες περιφερείας σε άλλα μέρη chassis-mounted που θα είχαν προκαλέσει αποτυχία.

Το BDS βασίζει τη διάγνωση του και στα 2 τις στρατηγικές του έμπειρου διαγνώστη και τη γνώση σχετικά με την κατασκευή, λειτουργία και τις τυχαίες σχέσεις των στοιχείων στην ηλεκτρονική συσκευή. Το BDS εκτελείται σε LES γλώσσα και χρησιμοποιεί μια rule-base αντιπροσώπευση σχήματος με

backward αλυσίδα. Αναπτύχθηκε στο Λόκχιντ Palo Alto Εργαστήριο Έρευνας και έφθασε στο στάδιο έρευνας πρωτοτύπων.

Το CADHELP παρακινεί έναν ειδικό να αποδείξει την λειτουργία γραφικών χαρακτηριστικών ενός σχεδίου subsystem βοηθούμενο από ένα computer-aided design subsystem (CAP) για σχεδιασμό digital λογικών κυκλωμάτων. Αυτό εξηγεί στον χρήστη πως να χρησιμοποιεί το CAD subsystem, επεξεργάζοντας τις εξηγήσεις για να ταιριάσει τις ανάγκες και επιθυμίες του χρήστη. Παρέχει εξηγήσεις στα Αγγλικά, όταν ο χρήστης κάνει λάθος ή ζητά βοήθεια και παράγει το κείμενο εξήγησης από τη βάση των γνώσεών του με τρόπο δυναμικό, βασιζόμενο σε γραπτά που συσχετίζονται με διαφορετικά χαρακτηριστικά του subsystem CAD. Καθώς ο χρήστης γίνεται πιο έμπειρος οι εξηγήσεις που δημιουργούνται από CADHELP γίνονται πιο σύντομες. Η γνώση στο σύστημα οργανώνεται σαν σειτ subsystem συνεργασίας ή "experts" που ελέγχονται από υψηλότερου επιπέδου έργα διευθυντού προγράμματος. Το σύστημα εκτελείται σε FRANZ LISP και τρέχει σε DEC VAX 11/780 κάτω από UNIX. Το CADHELP αναπτύχθηκε στο Πανεπιστήμιο του Connecticut και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το COMPASS αναλύει telephone switching systems μηνύματα διατήρησης για το GTE's No 2 EAX Switch και προτείνει δράσεις (ενέργειες) για να εκτελεσθεί.

Το σύστημα εξετάζει μηνύματα διατήρησης που περιγράφει καταστάσεις λάθους που συνέβηκαν κατά τη διάρκεια της λειτουργίας τηλεφωνικής συνδιαλέξεως του διακόπτη. Κατόπιν, αναγνωρίζει γκρουπ μηνυμάτων που πιθανόν προκαλούνται από κοινό λάθος, καθορίζει το πιθανά ειδικά σφάλματα στο διακόπτη και προτείνει πράξεις διατήρησης για να επικυρώσει και θεραπεύσει το σφάλματα.

Το σύστημα ενσωματώνει την πείρα ενός ειδικού στου διακόπτες (switch) και ολοκληρώνει τις γνώσεις για ατομική κατασκευή του διακόπτη, τα

σφάλματα του διακόπτη, τα μηνύματα διατήρησης και πιθανές ενέργειες διατήρησης. Το COMPASS εκτελείται σε KEE και INTER LISP-D για χρήση σε XEROX 1108 εργατικούς σταθμούς.

Αναπτύχθηκε από τα GTE εργαστήρια Inc. και έφθασε το στάδιο πρωτοτύπου.

Το CRTTER βοηθά τους μηχανικούς σχεδιασμού κυκλωμάτων να αναλύσουν την ορθότητα, την χρονικότητα, την ευρωστία και ταχύτητα των σχεδίων περιφερειαίας του VLSI. Το σύστημα δέχεται περιφέρειες σχηματικών και I/D ειδικεύσεις από τον μηχανικό και οικοδομεί ένα μοντέλο αντιλήψεως της περιφερειακής/κυκλικής παράστασης. Το CRITTER κάνει περίληψη αυτών των πληροφοριών και τις παρουσιάζει στον μηχανικό με μια εκτίμηση της περιφερειακής κατορθωτότητας τόσο καλά όσο τις διαγνωστικές και πληροφορίες επισκευής. Οι γνώσεις ενσωματωμένες μέσα στο CRITTER συμπεριλαμβάνουν πληροφορίες για διαγράμματα κυκλωμάτων και τεχνικές ανάλυσης κυκλωμάτων όπως: subcircuit simulation και path delay analysis. Τα περιφερειακά (κυκλικά) διαγράμματα αντιπροσωπεύονται χρησιμοποιώντας πλαίσια, ενώ άλλες γνώσεις είναι στον τύπο αλγεβρικών μορφών και κατηγορηματικών υπολογισμών. Το CRITTER εκτελείται με INTERLISP. Αναπτύχθηκε στο Ρούτζερς Πανεπιστήμιο κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το DAA βοηθά τους VLSI σχεδιαστές εκτελώντας μια hardware κατανομή από για αλογοριθμική περιγραφή ενός VLSI συστήματος. Το σύστημα παίρνει μια ροή περιγραφής δεδομένων ενός VLSI συστήματος και παράγει μια λίστα τεχνολογικών ανεξάρτητων καταγραφών, λειτουργιών, μονοπάτια δεδομένων και σήματα ελέγχου. Το expertise του συστήματος αποτελείται από αλογορίθμους για κατανομή hardware που συλλέγεται από έμπειρους σχεδιαστές τόσο καλά όσο η γνώση που βρίσκεται στον διανεμητή CMU/DA.

Το DAA είναι ένα forward chaining rule-based systems που εκτελείται δε OPS5. Αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το DFT βοηθά τους VLSI σχεδιαστές να ελέγχουν για DFT (design for testability) τους κανόνες παράβασης και να μετατρέπουν το σχέδιο για να τους απομακρύνουν. Μια φορά το σύστημα τοποθετεί στον DFT κανόνες παράβασης, αναλύει αυτούς που ονομάζεται (sensitive scan design method).

Το DFT επίσης αποσπά τον έλεγχο και την παρατήρηση πληροφοριών από την κατασκευή σχεδίου και χρησιμοποιεί αυτές για αυτόματη δοκιμαστική παραγωγή δείγματος. Οι γνώσεις στο σύστημα αντιπροσωπεύονται μέσα σε λογικά προσανατολισμένο πλαίσιο δουλειάς. Ένα digital design αποτελείται από σετ λογικών επιβεβαιώσεων που περιγράφουν τον κόμβο VLSI αλληλοσύνδεσης και τις λειτουργίες των κόμβων. Το DFT εκτελείται σε PROLOG. Αναπτύχθηκε στο Πανεπιστήμιο Syracuse και έφθασε στο στάδιο επίδειξης πρωτοτύπου.

Το EL εκτελεί μια σταθερή κατάσταση ανάλυσης των (resistor-diode-transistor circuits). Δίνοντας μια περιγραφή σχηματικού κυκλώματος το σύστημα αναλύει το κύκλωμα και καθορίζει τις αξίες διαφόρων circuit παραμέτρων όπως voltage, ή τρέχουσες τιμές σε δεδομένα σημεία.

Το EL's expertise συμπεριλαμβάνει γενικές αρχές των ηλεκτρονικών και χαρακτηριστικά περιφερειακών στοιχείων. Το σύστημα είναι rule-based και έχει forward chaining control scheme. Υπάρχει μια περιορισμένη εξήγηση που βασίζεται πάνω στην ιδιότητα ότι το EL θυμάται τις δικαιολογίες για νέους ισχυρισμούς. Το EL εκτελείται σε ARS. Αναπτύχθηκε στο MIT και έφθασε το στάδιο μιας έρευνας πρωτοτύπου.



Το EURISKO μαθαίνει νέα ευριστικά και καινούργιες κύριους-ειδικούς ορισμούς αντιλήψεων σε κυρίαρχο πρόβλημα. Το σύστημα μπορεί να μάθει με την ανακάλυψη αριθμού διαφορετικών κυρίων προβλημάτων, συμπεριλαμβανομένου του VLSI σχεδίου. Το EURISKO έχει καταπιαστεί με το πρόβλημα εφεύρεσης νέων ειδών τρισδιάστατων μικροηλεκτρονικών συσκευών που μπορούν να επινοηθούν/κατασκευαστούν, χρησιμοποιώντας λέιζερ τεχνικές ξανααποκρυστάλλωσης (recrystallization) και έχει σχεδιάσει καινούργιες και ενδιαφέρουσες μικροηλεκτρονικές συσκευές. Το EURISKO λειτουργεί παράγοντας για συσκευή διαμόρφωσης, (device configuration), υπολογίζοντας την συμπεριφορά του εκτιμώντας (διατιμώντας) την λειτουργικότητά του και κατόπιν αξιολογώντας αυτή ενάντια άλλων συγκρίσιμων συσκευών. Το σύστημα εκτελείται σε INTERLISP για XEROX 1100 series work stations. Αναπτύχθηκε στο Πανεπιστήμιο Stanford και είναι περισσότερο ένα AI πρόγραμμα για μάθηση με ανακάλυψη παρά ένα expert system.

Το FG502-TASP βοηθά τους τεχνικούς στη διάγνωση κακής λειτουργίας Tektronix FG502 function generators. Το σύστημα χρησιμοποιεί ad hoc κανόνες για σύστημα συμπεριφοράς μαζί με ευριστικά που συγκεντρώνονται από έμπειρους τεχνικούς μάλλον, παρά από θεωρητικό μοντέλο της FG502 λειτουργίας που θα απαιτούσε αρχές ηλεκτρονικών ανωτάτου επιπέδου. Επί πλέον με τις ερωτήσεις ο τεχνικός κατευθύνει τη διάγνωση, το σύστημα παρέχει μια γραφική επίδειξη των μερών σχεδίου σε σχετικό κυκλικό πίνακα και μια εικόνα του πως θα έπρεπε να μοιάζει το waveform όταν μετριέται σε δεδομένο σημείο στο πίνακα. Το FG 502-TASP είναι ένα object-oriented system που εκτελείται σε SMALLTA LK-80. Αναπτύχθηκε από την Tektronix και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το FOREST απομονώνει και κάνει διάγνωση σφαλμάτων σε ηλεκτρονικό εφοδιασμό. Το σύστημα συμπληρώνει στην αποφυγή λαθών και

την απομόνωση ικανοτήτων του τρέχοντος automatic test equipment (ATE) διαγνωστικού software. Οι γνώσεις του FOREST συμπεριλαμβάνει εμπειρικούς rules of thumb από ειδικούς μηχανικούς, γνώσεις για τη χρήση περιφερειακών διαγραμμάτων και γενικές troubleshooting principles.

Αυτή η γνώση κωδικοποιείται, χρησιμοποιώντας κανόνες σαν τους certainty factors του PROSPECTOR και ευκολίες σαν το explanation facility του MYCIN. Το FOREST εκτελείται σε PROLOG. Αναπτύχθηκε στο Πανεπιστήμιο της Πενσιλβανία σε συνεργασία με την RCA corporation και έφθασε το στάδιο της επίδειξης πρωτοτύπου.

Το IN-ATE βοηθά έναν τεχνικό να διορθώνει τα ελαττώματα σε Tectronix Model 465 oscilloscope αναλύοντας συμπτώματα και παράγοντας ένα δένδρο αποφάσεων από test points που θα ελεγχθούν από τον τεχνικό. Το σύστημα εφαρμόζει 2 τύπους κανόνων: εκείνους που παρέχονται από έναν expert διαγνώστη και εκείνους που παράγονται αυτόματα από ένα εσωτερικό μοντέλο του oscilloscope, που είναι ένα διάγραμμα μπλοκ της αυξανόμενης unit με στοιχεία βαθμών αποτυχίας. Η γνώση αντιπροσωπεύεται με rules οι οποίοι είναι εφοδιασμένοι με πιθανοτικούς συντελεστές βεβαιότητας. Καθώς οι κανόνες εφαρμόζονται, οι βεβαιότητες updated διαμέσου μιας procedure παρόμοιες με minimaxing. Το σύστημα εκτελείται σε FRANZ LISP. Αναπτύχθηκε στο Ναυτικό Εργαστήριο Έρευνας κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το MESSAGE TRACE ANALYSER βοηθά το debug real σε real time system. Τέτοια είναι μεγάλης τηλεπικοινωνίας switching μηχανές που περιέχουν εκατοντάδες processors. Το σύστημα εξετάζει τα ίχνη μηνυμάτων εσωτερικής πορείας, αναγνωρίζοντας παράνομες σειρές μηνυμάτων για να εντοπίσουν το σφάλμα μέσα σε μια πορεία. Το σύστημα θεωρεί την απεσταλμένη πορεία ID, της receiver process ID, τον τύπο μηνύματος και ratimestamp fields μηνυμάτων στο trace. Γενικά τα debugging heuristics και τα

facts για το ειδικό σύστημα που γίνονται debugged αντιπροσωπεύονται με την βοήθεια rules και εφαρμόζονται χρησιμοποιώντας και τα forward και backward chaining. Το σύστημα περιέχει ευκολίες που του επιτρέπει να απαντήσει σ'ερωτήσεις που αφορούν το reasoning. Το σύστημα γράφεται σε PROLOG. Αναπτύχθηκε στο Πανεπιστήμιο του Waterloo και έφθασε το στάδιο επίδειξης πρωτότυπου.

Το NDS ευρίσκει πολλαπλά λάθη σε εθνικό δίκτυο επικοινωνιών ονομαζόμενα COMNET εφαρμόζοντας εμπειρικές διαγνωστικές στρατηγικές βασιζόμενες σε γνώσεις για την τοπολογία δικτύων και έκθεσης (σύνθεσης). Το σύστημα προτείνει διαγνωστικά τεστ για να εκτελεστούν και το αποτέλεσμα κάθε τεστ παρέχει απόδειξη για ή ενάντια σε ένα σφάλμα που υπάρχει σε μερικά τεστ στοιχείων. Τα στοιχεία κάτω από αυτήν την θεώρηση συμπεριλαμβάνουν telecommunication processors modems telephone circuits και computer terminals.

Το NDS είναι ένα rule-base σύστημα και εκτελείται σε ARBY. Το σύστημα αναπτύχθηκε από τον Smart Systems Technology σε συνεργασία με την Εταιρία Shell Development κι έφθασε το στάδιο μιας επίδειξης πρωτοτύπου.

Το PALLADIO βοηθά τους circuit designers στο σχέδιο και δοκίμασμα των καινούργιων VLSI circuits. Το PALLADIO είναι ένα circuit design environment που συμπεριλαμβάνει “αλληλεπιδραστικούς” graphics editors που χειρίζονται υψηλού επιπέδου ηλεκτρονικά στοιχεία, έναν simulator ο οποίος με βάση την “κατασκευαστική” και “συμπεριφορική” specification of circuit επιτυγχάνει το simulate και μηχανισμούς για την διύλιση και δημιουργία design specifications σε διαφορετικά επίπεδα “αφαίρεσης”. Το PALLADIO έχει χρησιμοποιηθεί στο να σχεδιάζει μια ποικιλία των MOS circuits. Το σύστημα εκτελείται σε LOOPS που παρέχει object-oriented, rule-based και logic-oriented representation μηχανισμούς. Το PALLADIO αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο έρευνας πρωτότυπου.

Το PEACE βοηθά τους μηχανισμούς στο σχέδιο των electronic circuits. Το σύστημα είναι ένα CAP εργαλείο (tool) που εκτελεί ανάλυση και σύνθεση of pasive και digital circuits εφαρμόζοντας γνώση που αφορά το circuit design. Αυτή η γνώση συμπεριλαμβάνει λειτουργική περιγραφή βασικών circuits και δικτύων, λειτουργικές και τοπολογικές circuits transformations, τις στρατηγικές για ανάλυση και σύνθεση και heuristics για να προβλέπουν αποτυχίες. Το PEASE μπορεί να συνθέτει passive circuits από τις λειτουργικές τους περιγραφές και digital circuits από την μαθηματική έκφραση των transfer funtions. Το PEASE είναι ένα λογικά-προσανατολισμένο σύστημα και εκτελείται σε PROLOG. Αναπτύχθηκε στο Πανεπιστήμιο του Μάντσεστερ κι έφθασε το στάδιο μιας επίδειξης πρωτοτύπου.

Το REDESIGN βοηθά τους μηχανικούς στο redesign των digital circuits για να συναντήσουν τροποποιημένες functional specifications. Δίνοντας το στόχο του redesign το σύστημα 1) παράγει ευλογοφανείς local αλλαγές που θα δώσουν μέσα στο circuits βαθμούς στις αλλαγές που θα βασίζονται τόσο στο βαθμό δυσκολίας της εκτέλεσης όσο και στην ικανοποίηση στόχου και 2) ελέγχει για τα επιθυμητά πλάγια αποτελέσματα συσχετιζόμενα με τις αλλαγές. Το σύστημα παρέχει design assistance συνδέοντας τυχαία λογική αναλύοντας τις σχέσεις αιτίας-αποτελέσματος της circuit operation με functional reasoning και αναλύοντας τους σκοπούς ή ρόλους των circuits components. Η circuit knowledge στο Redesign αντιπροσωπεύεται σαν network of modules και data paths. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε στο στάδιο έρευνας πρωτοτύπου.

Το SADD βοηθά τους μηχανικούς στο design των digital circuits. Από τον μηχανικό το σύστημα δέχεται μια λειτουργική περιγραφή του προταθέντος circuits στα Αγγλικά και χρησιμοποιεί αυτό για να (κτίσει) οικοδομήσει ένα εσωτερικό μοντέλο of the circuit. Το σύστημα χρησιμοποιεί γνώσεις για το μοντέλο, component characteristics και circuit behavior με σκοπό να σχεδιάσει

ένα ευλογοφανή circuit το οποίο δοκιμάζει για ορθότητα κοινωνίας simulate στην λειτουργία του. Το SADD χρησιμοποιεί μια frame-based αντιπροσώπευση γνώσης που αφορά circuit components. Το σύστημα εκτελείται σε LISP. Αναπτύχθηκε στο Πανεπιστήμιο Maryland κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το SOPHIE διδάσκει τους μαθητές πως να διορθώνουν ελαττώματα σε electrical circuit fault (σφάλμα) επιτρέποντας στον μαθητή να συλλέγει ένα σφάλμα σε simulate circuit και μετά να πορεύεται στα βήματα για να βρει το σφάλμα. Σε κάθε βήμα το σύστημα ζητά από τον μαθητή μα μαντεύει και να περιγράψει την ποιοτική συμπεριφορά of the test instrument. Όταν ο μαθητής κάνει λάθος το σύστημα δείχνει την μέτρηση και δίνει τις απαραίτητες εξηγήσεις. Το σύστημα παρέχει μια τυπωμένη ιστορία of the diagnosis session για την ενημέρωση του μαθητή. The circuit simulator στο SOPHIE περιέχει ένα μηχανισμό για πρόπλαση και απεικόνιση τυχαίας διάδοσης σφάλματος π.χ. πως η “αποτυχία” (failure) μιας component μπορεί να γίνει αφορμή στο να αποτύχουν και άλλες components. Το SOPHIE εκτελείτε σε INTERLISP και FORTRAN, η τελευταία χρησιμοποιείται για το circuit simulator package. Το σύστημα αναπτύχθηκε στο Bolt, Beranek και Newman κι έχει εξελιχθεί σε 3 versions. Έφθασε το στάδιο έρευνα πρωτοτύπου.

Το SYN βοηθά τους μηχανικούς στο να συνθέτουν electrical circuits. Ο μηχανισμός εισάγει εν μέρει ειδικευμένα circuit διαγράμματα και περιορισμούς σε particular circuit components και το σύστημα συνδέει αυτές τις πληροφορίες με γνώσεις που αφορούν ενυπάρχοντες περιορισμούς στο circuit structure για να προδιαγράψει το circuit ολοκληρωτικά. Το σύστημα συνδυάζει περιορισμούς χρησιμοποιώντας symbolic algebraic manipulation of the formulas που περιγράφουν τις circuit components. Το σύστημα αναπτύχθηκε στο MIT και έφθασε το στάδιο έρευνα πρωτοτύπου.

Το TALIB αυτόματα συνδέει γραμμογραφήσεις ολοκληρωμένων κυκλωμάτων για MOS cells. Το σύστημα παίρνει σαν input 1) μια περιγραφή των circuit components που σχεδιάζονται επάνω σε μια κουλούρα από σιλικόνη 2) τις αλληλοσυνδέσεις τους και τις τοπολογικές και γεωμετρικές ανάγκες γύρω από το εξωτερικό σύνορο του circuit. Απ'αυτά το σύστημα παράγει ορθές και συμπαγείς γραμμογραφήσεις του cell. Το TALIB δημιουργεί και διωλίζει σχέδια για το σχεδιασμό του circuit και μετά εφαρμόζει τα σχέδια χρησιμοποιώντας γνώσεις που αφορούν subcircuit interconnection characteristics καθώς και περιορισμούς που υπάρχουν μεταξύ των subcircuits. Το TALIB είναι ένα forward chaining, rule-based σύστημα που εκτελείται σε OPS5. Αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon κι έφθασε στο στάδιο έρευνας πρωτοτύπου.

Το TRANSISTOR SIZING SYSTEM βοηθά τους σχεδιαστές των circuit παρέχοντάς τους το σχέδιο ολοκληρωμένων circuits. Αυτό το επιτυγχάνει εκτελώντας μέρος του υπάρχον μεταξύ των subcircuits. Το TALIB είναι ένα forward chaining, rule-based σύστημα που εκτελείται σε OPS5. Αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon κι έφθασε στο στάδιο έρευνας πρωτοτύπου.

Το TRANSISTOR SIZING SYSTEM βοηθά τους σχεδιαστές των circuit παρέχοντας τους το σχέδιο ολοκληρωμένων circuits. Αυτό το επιτυγχάνει εκτελώντας μέρος του ραφιναρίσματος (layout). Το σύστημα αποφασίζει το φυσικό μέγεθος των τρανζίστορ στο circuit θεωρώντας τους στόχους των σχεδιαστών που αφορούν θέματα ταχύτητας και κατανάλωσης δύναμης, συνδυασμένα με την γνώση της σχέσης ανάμεσα στην ταχύτητα και δύναμη για MOS circuits. Το Transistor sizing system αναλύει το circuit για να αποφασίσει τα critical paths και για παράγει a trade-off curve που βασίζεται σε simple delay models. Ο σχεδιαστής κατόπιν συλλέγει ένα goal point στην καμπύλη και το σύστημα resizes τα τρανζίστορ με το στόχο. Αυτό το transistor sizing system

χρησιμοποιεί ένα frame-base και object-oriented knowledge representation scheme. Εκτελείται σε LOOPS και ενσωματώνεται σε περιβάλλον του PALLADIO . Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Stanford και έφθασε το στάδιο έρευνας πρωτοτύπου.

## 5. ΜΗΧΑΝΟΛΟΓΙΑ

Το CONPHYDE βοηθά τους χημικούς μηχανικούς να συλλέγουν φυσικής ιδιότητας μεθόδους εκτίμησης. Το σύστημα χειρίζεται μια συλλογή vapor-liquid ισορροπίας συντελεστών για την δημιουργία process simulation, δίνοντας πληροφορίες για απαιτούμενη ακρίβεια και αναμενόμενες συγκεντρώσεις, θερμοκρασίες και σειρές πίεσης. Η γνώση στο CONPHYDE αντιπροσωπεύεται με rules και semantic nets όπως και στο PROSPECTOR. Τα inferences βασίζονται σε χρήση παραγόντων βεβαιότητας και Bayesian θεωρία απόφασης για propagating probabilities που σχετίζονται με τα δεδομένα. Το σύστημα εκτελείται σε KAS και χρησιμοποιεί την explanation facility του KAS. Το CONPHYDE αναπτύχθηκε στο Πανεπιστήμιο Carnegie Mellon και έφθασε στο στάδιο επίδειξης πρωτοτύπου.

Το DELTA βοηθά στην αναγνώριση και διόρθωση κακών λειτουργιών στις ντίζελ ηλεκτρικές ατμομηχανές εφαρμόζοντας διαγνωστικές στρατηγικές για την διατήρηση ατμομηχανής. Το σύστημα μπορεί να οδηγήσει τον χρήστη διαμέσου μιας ολόκληρης πορείας επισκευής, παρουσιάζοντας σχέδια μερών με τη βοήθεια κομπιούτερ και υποσυστήματα επισκευής σειρών στο τύπο videodisc movies και ειδική επισκευή οδηγιών όπου η κακή λειτουργία αναγνωρίζεται. Το DELTA είναι ένα rule-based system γραμμένο σε LISP . Το DELTA χρησιμοποιεί της forward και backward chaining καθώς και παράγοντες βεβαιότητας για να χειρισθεί “αβέβαιους κανόνες οικοδομής”. Αν

το σύστημα υλοποιήθηκε για πρώτη φορά σε LISP αργότερα ξαναεκτελέστηκε σε FORTH για (εγκατάσταση) installation on microprocessor-based systems. Η General Electric Company ανέπτυξε το σύστημα στο Schenectady της Νέας Υόρκης. Αυτοί συμπληρώνουν ένα field-testing του DELTA σαν προλούδιο στη χρήση του σε εμπορική βάση.

Το NPPC βοηθά τους nuclear power plant operators στο να καθορίσουν την αιτία κάποιου ανώμαλου γεγονότος εφαρμόζοντας κανόνες σε συνδυασμό με το plant operation. Το σύστημα χρησιμοποιεί ένα μοντέλο πρωταρχικά ψυχτικού συστήματος που περιλαμβάνει αντλίες, αντιδραστήρες, γεννήτριες ατμού και emergency core cooling system που κάνει διάγνωση στην αιτία μιας ανωμαλίας ή ατυχήματος και κατόπιν να προτείνει πορείες για να διορθώσει το πρόβλημα. Το μοντέλο αποτελείται από ένα commonsense algorithm network που προσχωρεί σε κατάλληλους διαγνωστικούς κανόνες. Το σύστημα αναπτύχθηκε στο Georgia Institute of Technology κι έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το REACTOR βοηθά τους reactor operators “χειριστές αντιδραστήρα” στην διάγνωση και θεραπεία των ατυχημάτων πυρηνικού αντιδραστήρα, εποπτεύοντας feedwater flow καθώς και επίπεδο περιορισμού ακτινοβολίας και ψάχνοντας για παρεκκλίσεις από φυσιολογικές συνθήκες λειτουργίας. “Όταν το σύστημα ανακαλύψει μια παρέκκλιση, αξιολογεί την κατάσταση και συστήνει κατάλληλη δράση, χρησιμοποιώντας γνώσεις για την διαμόρφωση του αντιδραστήρα και τις λειτουργικές σχέσεις των στοιχείων του μαζί με γνώσεις, σχετικά με την αναμενόμενη συμπεριφορά του αντιδραστήρα κάτω από γνωστές συνθήκες ατυχήματος. Το reactor εκτελείται σε LISP σαν ένα rule-based σύστημα και χρησιμοποιεί forward και backward chaining . Αναπτύχθηκε από EG&G και έφθασε το στάδιο παραγωγής πρωτοτύπου.



Το SACON βοηθά τους μηχανικούς να (καθορίσουν) στρατηγικές για ιδιαίτερα structural analysis problems. Οι μηχανικοί μπορούν κατόπιν να εκτελέσουν αυτή τη στρατηγική με την βοήθεια του MARK, το οποίο είναι ένα πρόγραμμα που χρησιμοποιεί finite-element analysis methods για να κάνει το simulation με την μηχανική συμπεριφορά των αντικειμένων. Το SACON αναγνωρίζει την τάξη ανάλυσης του προβλήματος και συστήνει ειδικά χαρακτηριστικά του MARK προγράμματος για να “θέσει” σ’ενέργεια όταν εκτελείται η ανάλυση. Χρησιμοποιεί γνώσεις για πιέσεις, εντάσεις και εκτροπές μιας κατασκευής κάτω από διαφορετικές συνθήκες φόρτωσης για να καθορίσει την κατάλληλη στρατηγική. Οι στρατηγικές που μπορούν να αναλυθούν περιλαμβάνουν φτερά αεροσκάφους, reactor pressure vessels, θήκες κινητήρων πυραύλου και γέφυρες. Το SACON είναι ένα backward chaining και rule-based σύστημα που εκτελείται σε EMYCIN. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το SPERIL-I εκτελεί κατασκευαστική διατίμηση καταστροφής των υπάρχουσών κατασκευών που υπόκεινται σε σεισμό. Δίνοντας επιταχυντόμετρα και visual inspection data το σύστημα καθορίζει την κατάσταση καταστροφής των κατασκευών. Του SPERIL-I το expertise αποτελείται από knowledge που συλλέγεται από “έμπειρους” και κάμψη. Οι γνώσεις αυτές αντιπροσωπεύονται σαν κανόνες και η μέθοδος εξαγωγής συμπερασμάτων είναι forward chaining. Το σύστημα χρησιμοποιεί παράγοντες βεβαιότητας συνδυασμένους με Fuzzy Logic για να υπολογίζει την καταστροφή τάξης μιας κατασκευής. Το SPERIL -I είναι γραμμένο σε γλώσσα C. Αναπτύχθηκε στο Rurdue στην Πανεπιστημιακή Σχολή των Πολιτικών Μηχανικών και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το SPERIL-II αξιολογεί την γενική ασφάλεια και την φθορά μιας υπάρχουσας κατασκευής. Το σύστημα αναλύει τα inspection data και τις instrument εγγραφές of structural responses κατά την διάρκεια ενός σεισμού,

όπως την επιτάχυνση και το εκτόπισμα σε ορισμένες τοποθετήσεις στην κατασκευή. Αυτό μετά αξιολογεί τα σχετικά ασφαλή χαρακτηριστικά και την θραυστότητα των στοιχείων της κατασκευής. Η γνώση του συστήματος προέρχεται από περίπτωση μελετών και αντιπροσωπεύεται υπό μορφή κανόνων κατηγορηματικής λογικής. Το SPERIL-II χρησιμοποιεί διάφορες μεθόδους λογικής, συμπεριλαμβανομένων και των δυο forward και backward chaining. Επίσης χρησιμοποιεί παράγοντες βεβαιότητας που συνδυάζονται χρησιμοποιώντας τον Dempster-Shafer algorithm. Πανεπιστήμιο Rurdue και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το STEAMER δίνει οδηγίες στους μαθητές Ναυτικής μηχανολογικής ώθησης πάνω στην λειτουργία ενός εργοστασίου ώθησης ατμού για μια 1078-class φρεγάτα. Το σύστημα μπορεί να προειδοποιήσει τον μαθητή να εκτελέσει the boiler light-off procedure για το εργοστάσιο, αναγνωρίζοντας τις κατάλληλες ενέργειες των μαθητών και διορθώνοντας τις ακατάλληλες. Το σύστημα δουλεύει δένοντας a mathematical simulation της ώθησης του εργοστασίου σε ένα graphical interface πρόγραμμα που επιδεικνύει ζωντανά χρωματιστά διαγράμματα των υποσυστημάτων του εργοστασίου. Οι μαθητές μπορούν να χειριστούν τις simulated components (τέτοιες βαλβίδες, μπρίζες) και να παρατηρούν τα αποτελέσματα στις παραμέτρους του εργοστασίου όπως αλλαγή στις πιέσεις, θερμοκρασίες και ροές. Το STEAMER εκτελείται σε ZETALISP και χρησιμοποιεί object-oriented representation scheme υποστηριζόμενο by ZETALISP'S FLAVORS package. Το σύστημα αναπτύχθηκε από το Naval Personal Research and Development Center σε συνεργασία με το Bolt, Bezanek and N

## 6. ΓΕΩΛΟΓΙΑ.

Το DIPMETER ADVISOR συμπεραίνει τις γεωλογικές κατασκευές υποεπιφανειών ερμηνεύοντας dipmeter logs και βρίσκοντας την αγωγιμότητα του βράχου. Το σύστημα χρησιμοποιεί γνώσεις που αφορούν τα dipmeter data και γεωλογία για να αναγνωρίζει χαρακτηριστικά στα dipmeter data και μετά τα συσχετίζει με την υπόγεια γεωλογική κατασκευή. Το σύστημα χρησιμοποιεί rules για την αναπαράσταση γνώσης και για εξαγωγή συμπερασμάτων το forward-chaining. Εκτελείται σε INTER-LISP-D και λειτουργεί στο Xerox 1100 workstations. Το σύστημα αναπτύχθηκε στο Schlumberger-Doll Research και έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το DRILLING ADVISOR βοηθά έναν oil-rig supervisor να αντιμετωπίζει και να αποφασίζει για την λύση των προβλημάτων που αφορούν τον μηχανισμό της διάτρησης και συγκεκριμένα να αντιμετωπίζει τις περιπτώσεις που υπάρχει κώλυμα σε διάτρημα κατά την διάρκεια της γεώτρησης. Το σύστημα κάνει διάγνωση στις πιο πιθανές αιτίες που κωλύματος (π.χ. κωνική τρύπα..) και συστήνει σετ θεραπειών για να λύσει το πρόβλημα και να λιγοστέψει την ευκαιρία να ξανασυμβεί. Το σύστημα βασίζει τις αποφάσεις του στη γνώση των γεωλογικών σχηματισμών στη τοποθεσία γεώτρησης και στη σχέση ανάμεσα στα παρατηρημένα συμπτώματα και στις ύποπτες αιτίες του προβλήματος. Η γνώση αντιπροσωπεύεται με την μορφή rules και η εξαγωγή συμπερασμάτων γίνεται με backward chaining. Το σύστημα αρχικά εκτελέσθηκε στο ΚΣ 300 και ξαναεκτελέσθηκε αργότερα σε S.1. Αναπτύχθηκε στο Teknowledge σε συνεργασία με την Societe Nationale Elf Aquitaine κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το ELAS δίνει συμβουλές για το πως να ελέγχεις κι ερμηνεύεις αποτελέσματα από το INLAN,(αυτό είναι ένα large-scale interactive program

για well log ανάλυση και επίδειξη αναπτυγμένο στο Amoco). Το ELAS βοηθά τον χρήστη: 1) συστήνοντάς του μεθόδους ανάλυσης, 2) προειδοποιώντας τον για τις ασυνεπείς directions of analysis και 3) δίνοντάς του περιλήψεις και ερμηνείες των αποτελεσμάτων από το user - INLAN interaction. Ο χρήστης κατευθύνει και τη μαθηματική ανάλυση του INLAN και την ερμηνευτική ανάλυση του ELAS, αλλάζοντας παραμέτρους ή επικαλούμενος tasks μέσω μιας sophisticated graphical display. Το σύστημα είναι rule-based και εκτελείται σε EXPERT. Αναπτύχθηκε στο Πανεπιστήμιο Rutgers σε συνεργασία με την AMOCO Production Research. Το ELAS έφθασε το στάδιο έρευνας πρωτοτύπου.

Το HYDRO βοηθά έναν υδρολόγο να χρησιμοποιεί το HSPF. Το HSPF είναι ένα computer program που "προσομοιάζεται" με την φυσική διεργασία που γίνεται κατά την διακοπή του precipitation μέσω watershed (κοιλιάς μεταξύ ποταμών). Το σύστημα βοηθά περιγράφοντας τα χαρακτηριστικά της watershed και δίνοντάς τα υπό την μορφή των αριθμητικών παραμέτρων στο HSPF. Ακολουθώντας εκτιμά αυτές τις παραμέτρους χρησιμοποιώντας γνώσεις για τον τύπο εδάφους, την χρήση της ξηράς, την βλάστηση και την γεωλογία. Το σύστημα is patterned μετά από το PROSPECTOR χρησιμοποιεί συνδυασμό rule-based και semantic net formalism για να κωδικοποιήσει τις γνώσεις του και βασίσει τα συμπεράσματα στη χρήση παραγόντων βεβαιότητας. Το σύστημα εκτελείται σε INTERLISP και αναπτύχθηκε από SR1 International. Το HYDRO έφθασε το στάδιο έρευνας πρωτοτύπου.

Το LITHO βοηθά τους γεωλόγους στο να ερμηνεύσουν δεδομένα από δρομόμετρα πηγών πετρελαίου (oil-well logs). Αυτά τα δεδομένα συμπεριλαμβάνουν καμπύλες (με την διαβίβασης ήχου και ραδιενέργειας). Το σύστημα χρησιμοποιεί τα log data Με τις γνώσεις της περιοχής γεωλογικού περιβάλλοντος (π.χ. γεωγραφίας, παλαιοντολογίας) για να χαρακτηρίσει την

πέτρα που συναντιέται σε ένα πηγάδι. Αυτός ο χαρακτηρισμός συνίσταται στα εξής στοιχεία: 1) Στοιχεία που αφορούν το πορώδες 2) Στοιχεία που αφορούν την υφή και τον τύπο των στρωμάτων. Το LITHO χρησιμοποιεί ένα πρόγραμμα ξεχωριστού δείγματος αναγνώρισης για να αποσπάσει χαρακτηριστικά απ' ευθείας από τα log data. Οι γνώσεις αντιπροσωπεύονται σαν κανόνες και η εξαγωγή συμπερασμάτων γίνεται με backward chaining. Το LITHO εκτελείται σε EMYCIN. Το σύστημα αναπτύχθηκε στο Schlumberger κι έφθασε το στάδιο μιας έρευνας πρωτοτύπου. (LITHOLof: περιγραφή των πετρωμάτων που βρίσκονται σε πηγάδια).

Το MUD βοηθά τους μηχανικούς να διατηρούν το optimal των drilling fluid properties. Αυτό γίνεται κάνοντας διάγνωση στις αιτίες των προβλημάτων των drilling fluids και προτείνοντας θεραπείες. Πιθανές αιτίες μπορεί να είναι: μολύνσεις (contaminants), υψηλές θερμοκρασίες ή πιέσεις και ανεπαρκή χρήση χημικών υλών. Το MUD περιέχει γνώσεις που αποσπώνται από τους domain experts και αφορούν drilling fluids καθώς και διαγνώσεις of drilling problems (προβλήματα γεώτρησης). Είναι ένα forward-chaining, rule-based σύστημα και χρησιμοποιεί MYCON-like certainty παράγοντες για να αντιπροσωπεύσει τις υποκειμενικές αποφάσεις των εμπειρογνομόνων. Επί πλέον, μπορεί να προμηθεύσει εξηγήσεις για τα σχέδια της συστηνόμενης θεραπείας. Το MUD εκτελείται σε OPS5. Αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon σε συνεργασία με το NL Baroid κι έφθασε το στάδιο field πρωτοτύπου. (Επειδή τα υγρά τους βρίσκονται από τις γεωτρήσεις έχουν θολή και λασπώδη εμφάνιση (muddy) γι' αυτό και το σύστημα ονομάστηκε MUD).

Το PROSPECTOR ενεργεί σαν σύμβουλος που βοηθά τους εξερευνητές γεωλόγους στην έρευνά τους για αποθέματα ορυκτά. Δίνοντας του field data που αφορούν για γεωλογική περιοχή αυτό εκτιμά την πιθανότητα που υπάρχει να βρεθούν ιδιαίτεροι τύποι μεταλλικών αποθεμάτων εκεί. Το σύστημα

μπορεί να διατιμήσει το δυναμικό για να βρει μια ποικιλία αποθεμάτων, συμπεριλαμβανομένων θειούχων νικελίου, ουράνιο αμμόπετρας, ανθρακούχου μολύβδου (carbonate lead/zink), κόκκινου χαλκού και αποθέματα κόκκινου (molybdenum). Το expertize του βασίζεται σε: 1) γεωλογικούς κανόνες που σχηματίζουν μοντέλα αποθεμάτων ορυκτών και 2) μια ταξινόμηση πετρωμάτων και μεταλλικών νερών. Το PROSPECTOR χρησιμοποιεί ένα συνδυασμό από rules και semantic nets για να αναπαραστήσει την γνώση του και βασίζει τα συμπεράσματά του στη χρήση των παραγόντων βεβαιότητας και την πρόπλαση των πιθανοτήτων που συσχετίζονται με τα δεδομένα. Το σύστημα εκτελέσθηκε στο INTERLISP και αναπτύχθηκε από το SR1 International. Το PROSPECTOR έφθασε το στάδιο παραγωγής πρωτοτύπου.

## 7. INFORMATION MANAGEMENT.

Το CARGuide βοηθά τους οδηγούς να βρουν πορείες και να οδηγηθούν σε δρόμους πόλεων. Το σύστημα χρησιμοποιεί τις τοποθεσίες αρχής και περιορισμού μαζί με τις πληροφορίες του χάρτη του και υπολογίζει μια optimum πορεία από το αρχικό σημείο στο περιορισμό. Η optimum πορεία ευρέσεως επιτυγχάνεται χρησιμοποιώντας ένα συνδυασμό του “διαίρει και βασίλευε” από precomputed πορείες και από τον Dijkstra’s shortest-path αλγόριθμο. Η πορεία επιδεικνύεται και δίδεται η έμφαση μέσω μιας γραφικής επίδειξης ενός χάρτη δρόμου. Κατά την διάρκεια του ταξιδιού η θέση του αυτοκινήτου κατά μήκος της πορείας είναι updated και displayed. Πριν από κάθε διασταύρωση, το σύστημα δίνει μια κατεύθυνση (ίσια, αριστερά ή δεξιά) καθώς και το όνομα του δρόμου που θα ακολουθηθεί. Η data base του χάρτη του δρόμου περιέχει πληροφορίες που αφορούν συσχετιζόμενα ονόματα δρόμων σε διασταυρώσεις και διασταυρώσεις σε

πορείες. Επίσης περιέχει pictorial πληροφορίες για παραγωγή εικόνων . Το CARGuide αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon κι έφθασε στο στάδιο επίδειξης πρωτοτύπου.

Το CODES βοηθά ένα data-base developer που θέλει να χρησιμοποιήσει την IDEF1 προσέγγιση για να ορίσει το conceptual schema μιας data base. Η γνώση στο CODES αντιπροσωπεύεται με την μορφή rules και η εξάγωση συμπερασμάτων γίνεται με backward chaining στρατηγική. Το CODES εκτελείται σε UCILISP. Αναπτύχθηκε στο University California κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το EDAAS βοηθά τους information specialist στο να αποφασίζουν ποιες πληροφορίες αφορούν την κατασκευή και διανομή τοξικών χημικών που μπορεί να απελευθερωθούν στο κοινό. Το σύστημα χρησιμοποιεί τις γνώσεις αξιολογώντας στο πότε οι πληροφορίες πρέπει να παρέχονται και στο πότε δεν μπορούν να δίνονται προς τα έξω. (Εδώ ανήκουν οι πληροφορίες που χαρακτηρίζονται σαν απόρρητες business information). Το EDAAS εκτελείται σε FORTRAN, αλλά χρησιμοποιεί επίσης και μια rule-based αναπαράσταση γνώσης, καθώς επίσης και ένα τροποποιημένο linear programming αλγόριθμο με την βοήθεια του οποίου λαμβάνονται αποφάσεις. Το σύστημα αναπτύχθηκε από τους Booz, Allen και Hamilton για χρήση από το Πρακτορείο Περιβαντολογικής Προστασίας (EPA) που το χρησιμοποιεί σε κανονική βάση. Το σύστημα έφθασε το στάδιο παραγωγής πρωτοτύπου.

Το FOLIO βοηθά τους διευθυντές χαρτοφυλακίου να καθορίζουν τους στόχους επένδυσης πελατών και να συλλέγουν χαρτοφυλάκια που εκπληρούν αυτούς τους στόχους. Το σύστημα καθορίζει τις ανάγκες πελατών κατά τη διάρκεια συνέντευξης και μετά συστήνει αναλογίες κάθε κεφαλαίου που παρέχουν ένα optimum fit για τους στόχους του πελάτη. Το FOLIO αναγνωρίζει ένα μικρό αριθμό τάξεων ασφαλειών και διατηρεί ένα σύνολο

γνώσεων που αφορούν τις ιδιότητες των ασφαλειών σε κάθε τάξη. Το σύστημα χρησιμοποιεί ένα forward chaining, rule-based σχήμα αναπαράστασης για να συμπεράνει στόχους (τελικούς σκοπούς) πελατών καθώς και ένα linear programming scheme για να μεγαλώσει το fit ανάμεσα στα goals και στο portfolio. Το FOLIO εκτελείται σε MRS. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το GCA βοηθά τους μεταπτυχιακούς φοιτητές να σχεδιάσουν their computer science curriculum. Το σύστημα μαζεύει πληροφορίες για την ακαδημαϊκή πορεία κάθε φοιτητή καθώς και τα ενδιαφέροντά του και κατόπιν ενεργεί σαν σύμβουλος προτείνοντας courses για το μαθητή. Το expertise του GCA συμπεριλαμβάνει - του Τμήματος και του Πανεπιστημίου - κανονισμούς που αφορούν : graduate degree programs, περιγραφές των courses προτιμούνται πιο πολύ από τους computer science students. Η γνώση στο GCA οργανώνεται υπό την μορφή 4 αλληλεπιδραστικών υποσυστημάτων τα οποία “διευθύνονται” από ένα managerprogram. Αυτά τα υποσυστήματα καθορίζουν: 1) τον αριθμό των μαθημάτων που θα έπρεπε να πάρει ο μαθητής 2) τα μαθήματα που ο μαθητής επιτρέπεται να πάρει 3) τα καλύτερα μαθήματα που παίρνει και 4) το καλύτερο πρόγραμμα για τον μαθητή. Του GCA η γνώση αναπαρίσταται με την μορφή rules οι οποίοι συνοδεύονται από παράγοντες βεβαιότητας. Το σύστημα εκτελείται σε PROLOG χρησιμοποιώντας a M-CIN-like inference engine. Αναπτύχθηκε στο Πανεπιστήμιο Duke κι έφθασε το στάδιο έρευνας πρωτοτύπου (Graduate Course Advisor).

Το IR-NLI παρέχει στους nontechnical users ένα natural language interface που αναφέρεται στις υπηρεσίες βελτίωσης πληροφοριών που προσφέρονται από on-line data bases. Το σύστημα ενεργεί σαν ένα front end δε διάφορες διαθέσιμες βάσεις δεδομένων και αποφασίζει ποιες θα είναι οι πιο κατάλληλες για να απαντήσουν στις απαιτήσεις του χρήστη. Το IR-NLI



συνδυάζει το expertise ενός επαγγελματία μεσολαβητή για on-line searching με την ικανότητα κατανόησης της φυσικής γλώσσας. Επίσης εκτελεί ένα διάλογο με το χρήστη. Η αντιπροσώπευση γνώσης γίνεται με rules που λειτουργούν πάνω σε 2 Knowledge bases. Η μία Knowledge base περιέχει domain-specific (DS) γνώση και η άλλη vocabulary γνώση ( VOC ). Η Knowledge base DS χρησιμοποιεί την τεχνική των semantic nets, για να ορίσει the data-base concepts και να δείξει πως αυτά σχετίζονται. Η Knowledge base VOC χρησιμοποιεί frames για να ορίσει το λεξικό of the application domain. Το IR-NLI εκτελείται σε FRANZ LISP. Αναπτύχθηκε στο Πανεπιστήμιο Udine, της Ιταλίας κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το PROJCON βοηθά έναν software development project διευθυντή, να κάνει διάγνωση των προβλημάτων του project και να βρει τις αιτίες τους. Το σύστημα συμβουλεύει τον διευθυντή του project και οικοδομεί ένα μοντέλο του ειδικού σχεδίου (specific project) και των προβλημάτων του. Όταν οι πιθανές αιτίες του προβλήματος έχουν καθορισθεί, το PROJCON επιδεικνύει την διάγνωση του κι εξηγεί τις λογικές πορείες του. Το PROJCON το expertise προέρχεται από project management expertw. Αυτή η γνώση υλοποιείται με την μορφή κανόνων οι οποίοι δίνουν τα συμπεράσματά τους μέσω ενός goal- directed και backward chaining inference mechanism. Το σύστημα εκτελείται σε EMYCIN. Αναπτύχθηκε στο Georgia Institute of Technology κι εκτελέσθηκε στο στάδιο κατορθωτής επίδειξης.

Το RABBIT βοηθά τους χρήστες να διατυπώσουν data-base queries δεδομένα. Αυτός ο έξυπνος data-base βοηθός, οδηγεί την διατύπωση των ερωτήσεων (queries) για χρήστες που έχουν μόνο μια αόριστη ιδέα του τι θέλουν. Η βελτίωση του expertise του συστήματος συνεπάγεται “βελτίωση με ξαναδιατύπωση(reformulation)”, όπου ο χρήστης κάνει ερώτηση κατασκευάζοντας μια εν μέρει περιγραφή των επιθυμητών data-base items. Το RABBIT εκτελείται σε smaltalk και λειτουργεί επάνω σε data bases που

αντιπροσωπεύονται in KL-ONE. Το σύστημα αναπτύχθηκε στο Xerox Palo Alto Research Center κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το RESEDA βοηθά ένα χρήστη να βελτιώσει τα βιογραφικά δεδομένα από τον κλάδο της Μεσαιωνικής Γαλλικής Ιστορίας. Ο χρήστης εισαγάγει το ερώτημά του που είτε ικανοποιείται by direct look up ή προκύπτει μέσω ενός search των πληροφοριών που βρίσκονται στην data base. (Το search επιτυγχάνεται με την βοήθεια inference procedures). Η data base των βιογραφικών δεδομένων, αποτελείται από frames που οικοδομούνται χρησιμοποιώντας μια μεταγλώσσα (metalanguage) βασισμένη σε ειδική γραμματική. Αυτή η βάση δεδομένων συμπεριλαμβάνει όλα τα γεγονότα που συνέβησαν τόσο στην ιδιωτική όσο και στη δημόσια ζωή των ανθρώπων που έχουν ενδιαφέρον για τους μελετητές της Γαλλικής Ιστορίας. Το expertise απαιτείται για την βελτίωση πληροφοριών που ενσωματώνονται υπό μορφή rules. Οι rules μπορούν να εκτελούν απλά συμπεράσματα ή μπορούν αυτόματα να εγκαθιδρύουν καινούργιους τυχαίους συνδέσμους ανάμεσα στα frames της knowledge base. Το RESEDA εκτελείται με τη γλώσσα VSAPL. Αναπτύχθηκε στο κέντρο Inter-Regional de Calcul Electronique (CIRCE) στη Γαλλία και έφθασε το στάδιο επίδειξης πρωτοτύπου.

## **8.LAW (NOMIKH.)**

Το AUDITOR βοηθά έναν επαγγελματία auditor (ελεγκτή) να αξιολογεί το δυναμικό του πελάτη για αθέτηση πληρωμής δανείου. Το σύστημα χρησιμοποιεί πληροφορίες για την ιστορία πληρωμής του πελάτη, την οικονομική κατάσταση, την πιστωτική κατάσταση και άλλες γνώσεις για να καθορίσουν εάν τα χρήματα θα πρέπει να κρατηθούν σαν απόθεμα για να καλύψει την παράληψη πληρωμής του δανείου του πελάτη. Το AUDITOR

είναι ένα rule-based σύστημα που εκτελείται σε AL/X, ένα παράγωγο του KAS. Λειτουργεί σαν μια εκδοχή του AL/X προσαρμοσμένο για χρήση in Pascal Systems on microcomputers. Το σύστημα αναπτύχθηκε στο University of Illinois Champaign-Urbana σαν ένα PhD κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το DSCAS βοηθά τους contractors να αναλύουν τις νόμιμες απόψεις από differing site condition απαιτήσεις (DSC). Μια DSC απαίτηση είναι μια χορηγούμενη θεραπεία για επιπρόσθετα έξοδα που γίνονται από έναν contractor, γιατί οι φυσικές συνθήκες διαφέρουν υλικά από εκείνες που δείχνονται στο αρχικό συμβόλαιο. Το DSCAS προμηθεύει έναν (contracting officer) συμβαλλόμενο υπάλληλο (CO) για μια δουλειά κατασκευής οικοπέδων (site) με το νόμιμο expertise που χρειάζεται για να χειρισθεί την DSC απαίτηση. Εάν το DSCAS καθορίζει ότι υπάρχει ένας λόγος για την μη συμπεριλαβή των ζητούντων επιπλέον εξόδων, η ανάλυση σταματά και δίδεται μια εξήγηση. Το DSCAS περιέχει ένα μοντέλο της decision process που χρησιμοποιείται από δικηγόρους για να αναλυθούν οι DSC απαιτήσεις. Αυτές οι γνώσεις αντιπροσωπεύονται υπό μορφή rules κανόνων. Η εξαγωγή συμπερασμάτων γίνεται με forward chaining. Το DSCAS εκτελείται σε ROSIE. Αναπτύχθηκε στο Πανεπιστήμιο του Colorado κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το JUDITH βοηθά τους δικηγόρους να συζητήσουν σχετικά με νομικές υποθέσεις πελατών. Το σύστημα ζητάει από τον δικηγόρο να του δώσει τις πραγματικές και νόμιμες υποθέσεις της αιτίας της "πράξεως". Αυτό μετά προτείνει επιπλέον προϋποθέσεις για θεώρηση του δικηγόρου, μέχρι να εξαντληθούν όλες οι σχετικές προϋποθέσεις για την περίπτωση. Η Knowledge base του JUDITH αποτελείται από υποθέσεις (premises) και construction files που δείχνουν τις σχέσεις που υπάρχουν ανάμεσα στα σύνολα των υποθέσεων (premises). Το JUDITH γράφεται σε FORTRAN. Αναπτύχθηκε στα

Πανεπιστήμια Heidelberg και Damstadt και είναι περισσότερο ένα AI περιβάλλον για εξερεύνηση legal reasoning παρά ένα πραγματικό Expert System.

Το LDS βοηθά τους εμπειρογνώμονες στην εκκαθάριση περιπτώσεων που έχουν παθητικό. Δίνοντας στο σύστημα μια περιγραφή της υπόθεσης που έχει παθητικό (product liability case) αυτό υπολογίζει το εναγόμενο παθητικό, την αξία της περίπτωσης και την αμερόληπτη εξόφληση του ποσού. Το expertise του βασίζεται τόσο στο τυπικό νόμιμο δείγμα όσο και στις ανεπίσημες αρχές και στρατηγικές των δικηγόρων καθώς και στις απαιτήσεις των διευθετών (adjustors). Το σύστημα υπολογίζει την αξία της περιπτώσεως, αναλύοντας το αποτέλεσμα: 1) από την αποτυχία (loss) : εδώ εξετάζονται οι ειδικές και γενικές ζημιές που προκύπτουν από το αδίκημα, 2) από το παθητικό: εδώ εξετάζεται η πιθανότητα του να αποδειχθεί το παθητικό του κατηγορούμενου, 3) από την ευθύνη (responsibility) : εδώ υπολογίζεται η “αναλογία” της κατηγορίας που εκχωρείται στους ενάγοντες για το αδίκημα, 4) από τα χαρακτηριστικά (characteristics): εδώ εξετάζονται οι υποκειμενικές θεωρήσεις όπως η επιδεξιότητα των δικηγόρων και η εμφάνιση των διαδίκων και 5) από το context : δίνονται οι θεωρήσεις που είναι βασισμένες στην στρατηγική, στον χώρο, και στον τύπο της απαιτήσεως. Το LDS είναι ένα rule-based σύστημα που εκτελείται σε ROSIE και αναπτύχθηκε από την Rand Cooperation. Το LDS έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το LEGAL ANALYSIS SYSTEM βοηθά τους δικηγόρους να εκτελούν απλές νόμιμες αναλύσεις που αφορούν σκόπιμα αδικήματα επιθέσεων και κτυπημάτων. Ο δικηγόρος εφοδιάζει το σύστημα με ένα σύνολο από facts τα οποία το σύστημα προσπαθεί να συνδέσει με σχετική νόμιμη δοξασία. Το σύστημα μετά παρουσιάζει τα συμπεράσματα του συμπεριλαμβάνοντας και την λογική με την οποία τα έβαλε. Παρέχει υποστήριξη για τα συμπεράσματα συμπεραίνοντας δικαστικές αποφάσεις και δευτερεύουσα νόμιμη εξουσία. Το

νόμιμο expertise , η δοξασία και τα case facts αντιπροσωπεύονται με την βοήθεια των semantic nets. Το σύστημα εκτελείται σε PLS (προκαταρκτική γλώσσα μελέτης). Αναπτύχθηκε στο MIT και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το LRS βοηθά τους δικηγόρους να ανακτήσουν πληροφορίες σχετικά με αποφάσεις δικαστηρίου και να ενημερωθούν πάνω στην νομοθεσία που αφορά μεταβιβάσεις τίτλων (π.χ. συναλλαγματικών) μια περιοχή εμπορικού νόμου που ασχολείται με επιταγές και ομόλογα. Το LRS περιέχει subjects περιγραφές που συνδέουν κάθε data item με την subject περιοχή που αναφέρεται σχετικά με αυτό. Υπάρχει ένα semantic net που περιέχει περισσότερα από 200 legal concepts. Για την βοήθεια του building χρησιμοποιήθηκαν 6 πρωταρχικά concepts (νόμιμο όργανο, ευθύνη, νόμιμη δράση, λογαριασμός, ποσό χρημάτων, και διάδικος) (legal instrument, liability, legal action, account, amount of money and party). Η γνώση στο LRS παρέχει σ'αυτό την ικανότητα να βγάλει συμπεράσματα για την "έννοια ερωτήσεων" που τίθενται. Το LRS αναπτύχθηκε στο Πανεπιστήμιο του Michigan και έφθασε το στάδιο πρωτότυπης έρευνας.

Το SAL βοηθά τους δικηγόρους και τους claims adjustors να αξιολογούν απαιτήσεις (claims) που σχετίζονται με "εκθέσεις ασβέστου". Το σύστημα χειρίζεται μια τάξη of diseases, την ασβέστωση και μια τάξη of plaintiffs (εναγόντων) τους απομονωτήρες. Το SAL παρέχει εκτιμήσεις στο πόσα χρήματα πρέπει να πληρώνονται οι ενάγοντες σε active υποθέσεις, βοηθώντας να προάγουν, άμεση τακτοποίηση. Το σύστημα χρησιμοποιεί γνώσεις που αφορούν: 1) καταστροφές, 2) ευθύνες κατηγορουμένου, 3) την υπευθυνότητα των εναγόντων και 4) χαρακτηριστικά υποθέσεων όπως ο τύπος των διαδίκων και η επιδεξιότητα των αντιθέτων δικηγόρων. Το SAL είναι ένα forward chaining, rule-based σύστημα που εκτελείται σε ROSIE . Το σύστημα

αναπτύσσεται στο Rand Corporating και έχει φθάσει το στάδιο επίδειξης πρωτοτύπου.

Το SARA βοηθά τους δικηγόρους να αναλύσουν αποφάσεις που διέπονται by discretionary norms. Δίνοντας τα γεγονότα (facts) μιας περίπτωσης και την ειλημμένη απόφαση σε έναν δικηγόρο, αυτός αναγνωρίζει παράγοντες (factors) θεωρημένους σχετικά με την απόφαση για παράδειγμα, το νοίκι που πληρώνεται για ένα διαμέρισμα, μπορεί να είναι σχετικός παράγοντας στο αν θα πρέπει να χορηγεί κοινωνική βοήθεια στο άτομο ή όχι. Ο δικηγόρος κατόπιν δίνει στο SARA τους παράγοντες factors (και τα αντίστοιχα values) που αφορούν μια ιδιαίτερη απόφαση. Στη βάση παραδειγμάτων σαν κι αυτά, το SARA απονέμει βάρη για κάθε παράγοντα (factor) προσαρμομένο ώστε να εξηγεί όσο το δυνατό περισσότερες από τις specified decisions. Οι παράγοντες που έχουν υψηλά βάρη θεωρούνται σπουδαίοι σε σχέση με τον discretionary norm κάτω από θεώρηση. Οι παράγοντες και οι αποφάσεις αντιπροσωπεύονται με την βοήθεια των frames. Επίσης για το weight calculation χρησιμοποιείται μια επαναληπτική μέθοδος συσχέτισης. Το SARA αναπτύχθηκε στο Norwegian Research Center for Computers and Law και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το TAXADVISOR βοηθά τους δικηγόρους που ασχολούνται με φορολογικές υποθέσεις πελατών που έχουν μεγάλη περιουσία (μεγαλύτερη από \$ 175000). Το σύστημα συλλέγει “δεδομένα πελατών” (client’s data) και συμπεραίνει ενέργειες που οι πελάτες είναι ανάγκη να ακολουθήσουν όσον αφορά τα οικονομικά τους. Επίσης εδώ συμπεριλαμβάνονται και συμβουλές που αφορούν ασφαλιστικές ενέργειες και συνταξιοδότηση, μεταφορά περιουσιακών στοιχείων και τροποποιήσεις για δωρεές και παροχές διαθήκης. Το TAXTDVISOR χρησιμοποιεί γνώσεις για το estate planning βασισμένες σε εμπειρίες και στρατηγικές δικηγόρων καθώς επίσης και αποδεκτές γνώσεις από text book . Το σύστημα χρησιμοποιεί ένα rule-based

σχήμα αντιπροσώπευσης γνώσεων ελεγχόμενο από backward chaining. Το TAXADVISOR εκτελείται σε ERYCIN . Αναπτύχθηκε στο Πανεπιστήμιο του Ιλινόις, Champaign-Urbana, σαν μια διατριβή Phd και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το TAXMAN βοηθά στην έρευνα νόμιμης λογικής νόμιμης επιχειρηματολογίας χρησιμοποιώντας the domain of corporate φορολογικού νόμου. Το σύστημα παρέχει ένα frame work για την αντιπροσώπευση των νόμιμων concepts και μια μεθοδολογία μετατροπών “transformation” αναγνώριση των σχέσεων ανάμεσα σ’εκείνα τα concepts. Μετασχηματισμοί (transformations) που γίνονται από την εξεταζόμενη περίπτωση σε άλλες συναφείς περιπτώσεις, δημιουργούν μια βάση για ανάλυση νόμιμης λογικής και επιχειρηματολογίας. Η γνώση που περιέχεται στο TAXMAN αντιπροσωπεύεται χρησιμοποιώντας frames και συμπεριλαμβάνει φορολογικές περιπτώσεις, νόμοι φορολογίας και αρχές για το transformation. Το TAXMAN I αρχικά χρησιμοποίησε ένα frame-like logical template representational formalism. Οι επόμενες versions απασχολούν ένα μοντέλο prototype-plus-deformation. Το TAXMAN εκτελείται σε AIMDS. Αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε το στάδιο έρευνας πρωτοτύπου.

## 9.MANUFACTURING (Κατασκευή)

Το IMACS βοηθά τους managers in a computer systems manufacturing environment with paperwork managment, capacity planning(σχεδιασμό χωρητικότητας). inventory management και άλλα έργα/καθήκοντα που σχετίζονται με το managing the manufacturing process. Το IMACS παίρνει ενός πελάτη την εντολή να παράγει ένα σχέδιο δύσκολο (ανώμαλο) στην

οικοδόμηση, από το οποίο μπορεί να εκτιμήσει τα προσόντα απαιτήσεων για παραγγελία. Πριν από το σύστημα του κομπιούτερ φτιαχθεί, το IMACS παράγει ένα λεπτομερές σχέδιο οικοδομής και χρησιμοποιεί για “monitor the computer system’s implementation”. Το IMACS είναι ένα forward chaining rule-based σύστημα οργανωμένο σαν ένα σύνολο of cooperating knowledge-based subsystems. Εκτελείται σε OPS5 . Το IMACS αναπτύχθηκε από την Digital Equipment Corporation κι έφθασε το στάδιο field prototype.

Το ISIS κατασκευάζει προγράμματα για δουλειά εργοστασίου. Το σύστημα συλλέγει σειρά λειτουργιών (operators) που χρειάζονται για να συμπληρώσουν μια παραγγελία, καθορίζει τις ώρες που αρχίζει και τελειώνει (τη δουλειά) και εκχωρεί resources σε κάθε μια λειτουργία (operator). Μπορεί επίσης να δράσει σαν έξυπνος βοηθός, χρησιμοποιώντας την πραγματογνωμοσύνη του για να βοηθήσει τους προγραμματιστές του εργοστασίου να διατηρούν συνέχεια του προγράμματος και να γνωστοποιήσει τις αποφάσεις εκείνες που καταλήγουν σε ανικανοποίητους περιορισμούς. Η γνώση στο σύστημα συμπεριλαμβάνει οργανωτικούς στόχους όπως πρέπουσες ημερομηνίες και κόστος, φυσικούς περιορισμούς, όπως limitations of particular machines και αιτιατούς περιορισμούς, όπως η σειρά κατά την οποία πρέπει να εκτελούνται οι λειτουργίες. Το ISIS χρησιμοποιεί frame-based knowledge σχήμα αντιπροσώπευσης, μαζί με κανόνες για την ανάλυση και επίλυση συμπλεκόμενων περιορισμών. Το σύστημα εκτελείται σε SRL. Αναπτύχθηκε στο Carnegie-Mellon University και δοκιμάσθηκε in the context of a Westinghouse Electric Corporation turbine component plant. Έφθασε στο στάδιο έρευνας πρωτοτύπου.

Το PTRANS βοηθά να ελέγξει τα manufacture and distribution of Digital Equipment Corporation’s computer systems. Χρησιμοποιεί τις περιγραφές και τις πληροφορίες που παρέχονται για την εντολή του πελάτη και αφορούν την δραστηριότητα του εργοστασίου για να αναπτύξει σχέδιο για συναρμολόγηση



και δοκιμή του παρελθόντος(order) computer system, συμπεριλαμβανομένου και του πότε θα φτιαχθεί το σύστημα. Το PTRANS monitors την πρόοδο των τεχνικών που εκτελούν το σχέδιο, κάνει διάγνωση των προβλημάτων, προτείνει λύσεις και προλέγει πιθανές επικείμενες ελλείψεις ή πλεονασμούς των υλικών. Το PTRANS σχεδιάστηκε για να δουλέψει με XSEL, ένα βοηθό πωλητού, έτσι ώστε όταν γίνεται η παραγγελία να μπορεί να επιβεβαιωθεί η ημερομηνία παράδοσης. Το PTRANS είναι ένα forward chaining , rule-based σύστημα που εκτελείται σε OPS5. Αναπτύχθηκε από την Digital Equipment Corporation σε συνεργασία με το Carnegie-Mellon University και έφθασε το στάδιο πρωτότυπης έρευνας.

## 10.ΜΑΘΗΜΑΤΙΚΑ.

Το ADVISOR βοηθά τους αρχάριους χρήστες του MACSYMA στο να κάνουν διάγνωση στις παραινέσεις (παρεξηγήσεις) και τους παρέχει συμβουλές ανάλογα με τις ανάγκες του κάθε χρήστη. Ο χρήστης δίνει στο ADVISOR μια σειρά user-entered of the overall goal Macsyma commands και μια έκθεση (περιγραφή) που δεν συναντήθηκε στην σειρά των commands. Το συμπεραίνει το σχέδιο του χρήστη και μετά παράγει συμβουλές. Το ADVISOR συνδυάζει πραγματογνωμοσύνη για τη χρήση του Macsyma μαζί ένα μοντέλο της συμπεριφοράς του αρχάριου χρήστη, που βοηθά το ADVISOR να αναγνωρίσει τα σχέδια και τις παρενοήσεις του χρήστη. Τα user plans και τα goals αντιπροσωπεύονται σαν data-flow graphs και goal trees. Το ADVISOR γράφτηκε σε MACLISP στο MIT κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το MACSYMA εκτελεί συμβολικό χειρισμό αλγεβρικών εκφράσεων και χειρίζεται προβλήματα που περιλαμβάνουν υπολογισμούς ορίων symbolic

integration, λύση εξισώσεων, canonical simplification και pattern matching. Το σύστημα χρησιμοποιεί μαθηματική πραγματογνωμοσύνη οργανωμένη υπό την μορφή ατομικών Knowledge sources και διαλεγμένη για ένα ειδικό πρόβλημα by sophisticated pattern-matching. Το MACSYMA επιτυγχάνει πολύ υψηλή ποιότητα και αποτελεσματική εκτέλεση στα μαθηματικά προβλήματα με τη σκοπιά του. Το σύστημα εκτελέσθηκε με LISP και αναπτύχθηκε κάτω από Project Mac στο MIT. Το MACSYMA έχει φθάσει το στάδιο εμπορικού συστήματος και χρησιμοποιείται κανονικά από μηχανολόγους και επιστήμονες σ'όλη την Αμερική.

Το MATHLAB 68 βοηθά τους μαθηματικούς, τους επιστήμονες και μηχανολόγους με το συμβολικό Αλγεβρικό χειρισμό που συναντιέται in analysis problems. Το σύστημα εκτελεί διαφορετικά polynomial factorization, αόριστη ολοκλήρωση καθώς ευθείς και αντίστροφους μετασχηματισμούς Laplace. Επίσης λύνει γραμμικές διαφορετικές εξισώσεις με constant symbolic coefficients. Περιέχει μαθηματική πραγματογνωμοσύνη οργανωμένη σε ατομικά modules καθένα από τα οποία έχει ιδιαίτερη λειτουργική ικανότητα. Τα δεδομένα των χρηστών ταξινομούνται σε 3 κατηγορίες, εκφράσεις εξισώσεις και functions. Οι κανόνες (rules) για τον αλγεβρικό χειρισμό των δεδομένων ποικίλλουν σε κατηγορίες. Το σύστημα εκτελέσθηκε με LISP και σχημάτισε τον γωνιαίο λίθο για την ανάπτυξη του MACSYMA. Το MACSYMA ανεπτύχθη στο MIT και έφθασε το στάδιο έρευνας πρωτοτύπου.

## 11. MEDICINE (Ιατρική)

Το ABEL βοηθά τον κλινικό ιατρό στην διάγνωση των acid-base και electrolyte διαταραχών που εμφανίζονται στους ασθενείς, εφαρμόζοντας γνώση που αφορά τις ασθένειες και τα συμπτώματα που τις παράγουν. Το σύστημα χρησιμοποιεί ένα τυχαίο μοντέλο των πιθανών ασθενειών του ασθενή για να θέσει ερωτήσεις στον κλινικό γιατρό και για να οδηγήσει τη διαγνωστική λογική πορεία (diagnostic reasoning process). Αυτό το μοντέλο περιέχει δεδομένα (data) για τον ασθενή καθώς και γνώση για τις σχέσεις ανάμεσα στις διάφορες “φάσεις” της ασθένειας. Το σύστημα ελέγχει για εσφαλμένα δεδομένα συγκρίνοντας τις απαντήσεις που αφορούν ερωτήματα που έχουν τεθεί με τις προσδοκούμενες που παράγονται από το μοντέλο. Η γνώση αντιπροσωπεύεται μέσα σένα τυχαίο network. (Αυτό είναι τύπου σημαντικού δικτύου και ειδικεύει τις σχέσεις αιτίας-αποτελέσματος μεταξύ diseases and findings). Το σύστημα αναπτύχθηκε σε MIT και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το AI/COAG βοηθά τους γιατρούς στο να κάνουν διάγνωση σε ασθένειες αιμόστασης. Η βασική τους βοήθεια είναι η ανάλυση κι ερμηνεία κλινικών τεστ εργαστηρίου για την πήξη του αίματος. Το σύστημα χειρίζεται 6 τεστ πήξης αίματος επί οθόνης συμπεριλαμβανομένων και εκείνων για platelet count και διαλυτότητα πήξεως ουρίας. Επιπλέον, το σύστημα μπορεί να αξιολογήσει το κλινικό ιστορικό αιμόστασης του ασθενούς για να επιβεβαιώσει τη διάγνωση που προτείνεται από τα τεστ της οθόνης. Το AI/COAG αναπτύχθηκε στο Πανεπιστήμιο στη Σχολή Ιατρικής του Μισσούρι και εκτελέσθηκε σένα μικροκομπιούτερ DECLSI-11. Έφθασε το στάδιο έρευνας πρωτοτύπου.

Το AI/MM αναλύει την συμπεριφορά στην περιοχή της νευρικής φυσιολογίας και εξηγεί τον “αιτιολογικό” του (rasionale) για τις αναλύσεις

του. Το σύστημα απαντά σε ερωτήσεις που αφορούν τις αξίες διαφόρων παραμέτρων, όπως ο όγκος του νερού που υπάρχει στο σώμα και ερμηνεύει παρατηρήσεις, όπως η εισαγωγή πολύ νερού ανώμαλα. Του ΑΙ/ΜΜ η πραγματογνωμοσύνη (expertise) περιλαμβάνει νόμους της φυσικής και ανατομίας, ιδρυτικές/θεμελειώδεις αρχές φυσιολογίας και εμπειρική γνώση για φυσιολογικές πορείες. Αυτή η γνώση αντιπροσωπεύεται με την βοήθεια rules που συμπεριλαμβάνουν ένα λεπτομερές τυχαίο μοντέλο. Κάθε κανόνας επίσης περιέχει μια περιγραφή της βασικής αρχής για χρήση στο να εξηγεί τη λειτουργία του συστήματος. Το ΑΙ/ΜΜ εκτελείται σε ΜRS. Αναπτύχθηκε στο Πανεπιστήμιο Stanford και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το ΑΙ RHEUM βοηθά τους γιατρούς στο να κάνουν διάγνωση σε ασθένειες συνδετικών ιστών κλινικής ρευματολογίας εφαρμόζοντας τυπικά διαγνωστικά κριτήρια που τα παίρνουν από εμπειρογνώμονες ρευματολόγους. Το σύστημα χρησιμοποιεί τα συμπτώματα ασθενούς και laboratory findings για να είναι ικανό να παρέχει βοήθεια σε 7 ασθένειες, καθώς επίσης και στις ρευματοειδή αρθρίτιδα, προοδευτική σκλήρωση του συστήματος και ασθένεια του Sjogren. Το ΑΙ RHEUM είναι ένα rule-based σύστημα που εκτελείται σε EXPERT και έχει access στους κανόνες του διαμέσου forward chaining. Αναπτύχθηκε στο Πανεπιστήμιο Missouri στην Ιατρική Σχολή κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το ANGY βοηθά τους γιατρούς να κάνουν διάγνωση στην στένωση στεφανιαίων αγγείων, αναγνωρίζοντας και απομονώνοντας τα στεφανιαία αγγεία σε angiograms (αγγειογράμματα). Το σύστημα πρώτα processes digital angiograms των στεφανιαίων αγγείων για να αποσπάσει την αρχική γραμμή και τα χαρακτηριστικά της χώρας. Αυτές οι πληροφορίες κατόπιν δίνονται στον ANGY το knowledge-based expert υποσύστημα που απευθύνονται στις low and high-level stages of vision processing. Το low level image processing stage χρησιμοποιεί rules για διαμέριση grouping and shape

analysis για την ανάπτυξη των αρχικών γραμμών και περιοχών μέσα σε πιο σημαντικά αντικείμενα. Το στάδιο high-level χρησιμοποιεί γνώσεις της καρδιακής ανατομίας και φυσιολογίας για να ερμηνεύσει το αποτέλεσμα να αναγνωρίσει τις σχετικές κατασκευές (π.χ.. την αορτή) και να εξαφανίσει άσχετες κατασκευές ή τεχνουργήματα που προκαλούνται by noise. Οι low-level image processing routines εκτελούνται στη C και συμπεριλαμβάνουν the edge detector και τον αναπτυχτή της περιοχής. Η ιατρική πραγματογνωμοσύνη αντιπροσωπεύεται υπό μορφή κανόνων που εκτελούνται με OPS5 και LISP. Το ANGY αναπτύχθηκε στο Πανεπιστήμιο της Pennsylvania κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το ANNA βοηθά τους γιατρούς στο να χορηγήσουν δακτυλιδίνη (φάρμακο) σε ασθενείς καρδιακής παθήσεως προβλήματα, όπως αρρυθμία και καρδιακή συμφόρηση. Το σύστημα χρησιμοποιεί συμπτώματα ασθενών και ιστορικά για να καθορίσει την κατάλληλη δόση δίαιτας καθώς επίσης και το ποσόν της δακτυλιδίνης που θα πρέπει να δοθεί στους ασθενείς και την αναλογία κατά την οποία θα έπρεπε να παρθεί (ληφθεί). Το σύστημα διατάσσει μια αρχική δόση, προειδοποιεί για την αντίδραση του ασθενή στο ναρκωτικό προσαρμόζοντας τη δόση κατάλληλα όταν η αντίδραση του ασθενούς δεν είναι η ίδια με την προσδοκώμενη αντίδραση. Το ANNA εκτελείται σε LISP. Αναπτύχθηκε στο MIT και έφθασε το στάδιο έρευνας πρωτοτύπου.

Το ARAMIS βοηθά τους γιατρούς να προσδιορίζουν καινούργιους ασθενείς με ρευματικές παθήσεις. Το σύστημα βρίσκει τα data(δεδομένα) που αφορούν προηγούμενες ρευματικές ασθένειες ασθενών και εκτελεί στατιστικές αναλύσεις πάνω σ'αυτά τα δεδομένα. Βασισμένο πάνω σ'αυτή την δραστηριότητα, το ARAMIS προσφέρει μια προγνωστική ανάλυση, συστήνει θεραπεία και παράγει μια prose case analysis. Η γνώση στο ARAMIS αποτελείται 1) από μια συλλογή στατιστικών μεθόδων ανάλυσης και 2) από

έναν αριθμό data-bases που περιέχουν λεπτομερή ιστορικά του ασθενούς. Τα δεδομένα του ασθενούς αποθηκεύονται σε βάση δεδομένων επί πίνακος χρησιμοποιώντας TOP, (Time oriented Databank System). Αυτό το system ακολουθεί σχετικές κλινικές παραμέτρους over time (πάνω από την κανονική ώρα). Το ARAMIS γράφτηκε σε PL\X, έναν ειδικευμένο τύπο του PL\T. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο field prototype.

Το ATTENDING δίνει οδηγίες σε φοιτητές Ιατρικής της αναισθησιολογίας κριτικάροντας τα σχέδια των φοιτητών για anesthetic management. Το σύστημα παρουσιάζει στον φοιτητή έναν υποθετικό ασθενή έτοιμο να υποβληθεί σε εγχείρηση και αναλύει τα management plan που σκοπεύει να εφαρμόσει ο φοιτητής. Το σύστημα βασίζει την ανάλυσή του σε μια αξιολόγηση κινδύνων που εμπλέκονται στη συνάφεια των ιατρικών προβλημάτων του ασθενούς. Η κριτική που παράγεται από το σύστημα έχει τον τύπο των σχολίων, τυπικά 4 ή 5 παραγράφους του Αγγλικού κειμένου. Το ATTENDING αναπτύχθηκε στο Πανεπιστήμιο του Yale την Ιατρική σχολή κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το BABY βοηθά τους κλινικούς γιατρούς by monitoring ασθενείς σε μια (NMEΦ) νεοσυσταθείσα μονάδα εντατικής φροντίδας. Το σύστημα προσπαθεί να βρει κλινικά σπουδαία patters ανάμεσα στα medical και demographic data που αφορούν τους ασθενείς της NMEΦ. Εποπτεύει όλα τα On-line δεδομένα στη NMEΦ, κρατάει ίχνη (track) των κλινικών καταστάσεων των ασθενών, προτείνει περαιτέρω αξιολόγηση για σπουδαία ευρήματα κι απαντά σε ερωτήσεις σχετικά με τους ασθενείς. Το BABY περιέχει neonatology, ιατρική πραγματογνωμοσύνη και την χρησιμοποιεί για να ερμηνεύσει a clinical και demographic data. Το BABY είναι ένα forward chaining, rule-based σύστημα (εδώ χρησιμοποιήθηκε στο PROSPECTOR). Το BABY γράφεται σε PASCAL και τοποθετείται στο ADVISE σύστημα, ένα ολοκληρωμένο, rule-based software περιβάλλον. Αναπτύχθηκε στο

Πανεπιστήμιο του Ιλινόις, Champaign-Urbana κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το BLUEBOX συμβουλεύει έναν γιατρό στο να βρίσκει την κατάλληλη θεραπεία για ασθενείς που πάσχουν από κατάθλιψη. Το σύστημα χρησιμοποιεί γνώσεις για τα συμπτώματα του ασθενούς και πληροφορίες που αφορούν την σωματική (από άποψη ιατρικής) την ψυχική κατάσταση του ασθενούς, καθώς και παρατηρήσεις για το αν είχε ή έχει πάρει ναρκωτικά και φυσικά παρατηρήσεις που αφορούν τόσο το ιστορικό του ασθενούς όσο και της οικογένειάς του. Με την βοήθεια αυτών γίνεται η διάγνωση της αθυμίας και προτείνεται ένα σχέδιο θεραπείας γι'αυτή. Αυτό το σχέδιο συμπεριλαμβάνει αποφάσεις για την νοσηλεία και θεραπεία για τους ασθενείς που κάνουν χρήση ναρκωτικών. Το BLUEBOX είναι ένα rule-based σύστημα που εκτελείται σε EMYCIN και αναπτύχθηκε στο Πανεπιστήμιο Stanford. Το σύστημα δοκιμάσθηκε στο περιβάλλον of the Palo Alto VA Mental Health Clinical Research Center κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το CASNET\GLAUCOMA κάνει διάγνωση σε καταστάσεις που συσχετίζονται με το γλαύκωμα και διατάσσει σχέδια ή θεραπείες για να το θεραπεύσει. Το σύστημα βασίζει τις αποφάσεις του στη γνώση που αφορά τις σχέσεις μεταξύ συμπτωμάτων ασθενούς, αποτελέσματα τεστ, εσωτερικές ανώμαλες συνθήκες, διάφορες φάσεις ασθένειας και σχέδια θεραπείας. Το σύστημα παρέχει μια περιγραφική ερμηνεία της κάθε περίπτωσης και μπορεί να retrieve literature για να υποστηρίξει τα συμπεράσματά του. Οι γνώσεις αντιπροσωπεύονται με έναν ιδιαίτερο τύπο σημαντικού δικτύου γνωστό σαν δίκτυο αιτιολογικής σχέσης (causal-association networks) Το σύστημα εκτελείται σε FORTRAN και αναπτύχθηκε στο Πανεπιστήμιο Rutgers. Έφθασε το στάδιο έρευνας πρωτοτύπου.

Το CENTAUR βοηθά τους πνευμονο-φυσιολόγους στην διαγνωστική ερμηνεία των τεστ πνευμονικής λειτουργίας. Το σύστημα χρησιμοποιεί μετρήσεις του ποσοστού αέρος στους πνεύμονες και τους βαθμούς ροής αερίων μέσα και έξω από τους πνεύμονες για να καθορίσει την παρουσία και αυστηρότητα της ασθένειας του πνεύμονα στον ασθενή. Η πραγματογνωμοσύνη που περιέχεται στο CENTAUR συμπεριλαμβάνει πνευμονική φυσιολογία και πρωτοτυπικά αποτελεσμάτων τεστ των πνευμόνων για κάθε πνευμονική ασθένεια. Η αναπαράσταση της γνώσης γίνεται με συνδυασμό frames και rules και certainty measures παρόμοια με εκείνες του EMYCIN certainty factors. Αυτοί χρησιμοποιούνται για να αποδείξουν πόσο πολύ τα πραγματικά δεδομένα ταιριάζουν με τα αναμενόμενα δεδομένα των αξιών του πρωτοτύπου. Επί πλέον, το CENTAUR έχει μια ικανότητα εξήγησης που παρέχει δικαιολογίες από ένα σημαντικό μοντέλο (σχέδιο) της πνευμονικής φυσιολογίας. Το CENTAUR εκτελείται σε INTERSLIP. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο μιας έρευνας πρωτοτύπου.

Το CLOT βοηθά τους γιατρούς να αποφασίσουν για τις ανωμαλίες σε σχέση με την πήκτικότητα του αίματος. Το σύστημα κάνει διάγνωση σε περίπτωση αιμορραγίας αναγνωρίζοντας ποιο από τα δύο υποσυστήματα πήξεως, το platelet-vascular ή το coagulative μπορεί να είναι προβληματικό. Η πρωταρχική ώθηση για την εκτέλεση του CLOT ήταν να μελετήσει τα εργαλεία της απόκτησης γνώσης και τις τεχνικές, έτσι η ιατρική του πραγματογνωμοσύνη δεν ήταν ραφιναρισμένη ή πλήρης δοκιμασμένη. Το CLOT είναι ένα backward chaining, rule-based σύστημα που εκτελείται σε EMYCIN. Το σύστημα κάνει χρήση της γνώσης που αποκτάται από το EMYCIN. Και χρησιμοποιεί τους ίδιους certainty factor mechanisms. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο επίδειξης πρωτοτύπου.





Το DIAGNOSER βοηθά τους γιατρούς ν'αναγνωρίζουν τις εκ γενετής καρδιακές παθήσεις και ειδικά την καρδιακή ανωμαλία, την γνωστή σαν total anomalous pulmonary venous connection. Το σύστημα εσκόπευε να βοηθήσει τους ερευνητές τεχνητής ευφυΐας να αναπτύξουν και να κάνουν "προφητικά" τεστ για τη φύση των λαθών με diagnostic reasoning. Παρουσιάζόμενα με στηθοσκόπηση και X-ray data το DIAGNOSER αποφασίζει (καθορίζει) την διάγνωσή του με υποθετικές ασθένειες των οποίων το πρωτότυπο ταιριάζει στις παρατηρήσεις. Το DIAGNOSER περιέχει γνώσεις σχετικά με diagnostic reasoning καθώς επίσης και γνώσεις που αφορούν την καρδιακή φυσιολογία, ανατομία και παθοφυσιολογία των αιτιολογικών κατασκευών που είναι σημαντικές για τις εκ γενετής καρδιακές παθήσεις. Η γνώση που αφορά την ασθένεια αντιπροσωπεύεται με την βοήθεια των frames. Η diagnostic reasoning γνώση αντιπροσωπεύεται με την βοήθεια rules οι οποίοι είναι ενσωματωμένοι σε πλαίσια. Το DIAGNOSER εκτελείται σε LISP 1.4. Αναπτύχθηκε στο Πανεπιστήμιο της Minnesota κι έφθασε το στάδιο έρευνας πρωτύπου.

Το DIALYSIS THERAPY ADVISOR βοηθά τους γιατρούς να επιλέξουν μια αρχική dialysis regimen για έναν ασθενή έτοιμο να αρχίσει θεραπεία διαρκούς αιμοδύαλσης. Στο σύστημα δίνεται το φύλο του ασθενούς, το ύψος, το βάρος, το BUN, το βάρος ουρίας και η συγκέντρωση αζώτου στα ούρα και αυτό παράγει έναν κατάλογο αποδεκτών θεραπειών. Η πραγματογνωμοσύνη στο σύστημα, αποτελείται από κανόνες που ανταποκρίνονται στον reasoning των εμπειρογνομόνων ιατρών, on how to specificity initial hemodialysis συνταγές. Η Εξαγωγή συμπερασμάτων από τους rules γίνεται μέσω forward και backward chaining. Το σύστημα ενσωματώνει equation solving explanation handling data-base retrieval μηχανισμούς, metarules για planning και μια ευκολία επικοινωνίας για την επαφή του με τον χρήστη. Το DIALYSIS THERAPY ADVISOR εκτελείται σε FRANZ LISP.

Αναπτύχθηκε στο Πανεπιστήμιο Vanderbilt κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το DIGITALIS ADVISOR βοηθά τους γιατρούς με το να συστήνει κατάλληλη θεραπεία digitalis για ασθενείς με καρδιακή συμφόρηση ή μεταβιβαστικές ενοχλήσεις της καρδιάς. Το σύστημα ρωτά τον κλινικό γιατρό, σχετικά με το ιστορικό του ασθενούς. π.χ. ηλικία, καρδιακός ρυθμός, επίπεδο ορού καλίου κ.λ.π. και κατόπιν παράγει σερ συστάσεων για αρχική θεραπεία. Αφού ο ασθενής έχει λάβει μια αρχική δόση το πρόγραμμα αναλύει την αντίδραση του ασθενούς, όπως δίνεται από τις απαντήσεις του κλινικού με άλλο σερ ερωτήσεων και παράγει καινούργια δόση δίαιτας (dosage regimen). Η πραγματογνωμοσύνη της digitalis therapy αντιπροσωπεύεται σαν μια ιεραρχία of concepts σε ένα σημαντικό δίκτυο (semanticnet). Ένας interpreter εκτελεί plans στην knowledge base (π.χ.. ελέγχει για digitalis ευαισθησία, ανάλογα με την αύξηση της ηλικίας). Το σύστημα περιέχει μια explanation facility που παράγει εξηγήσεις απ'ευθείας από τον executed code. Το DIGITALIS ADVISOR εκτελείται σε OWLI. Αναπτύχθηκε στο MIT και έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το DRUG INTERACTION CRITIC βοηθά τους γιατρούς να αποφασίσουν (how to administer drugs in the presence of other drugs) πως να διαχειρισθούν ναρκωτικά με την παρουσία άλλων ναρκωτικών. Το σύστημα αναγνωρίζει και τις ενάντιες (αρνητικές) και τις ενεργητικές αλληλεπιδράσεις, εξηγεί γιατί συμβαίνουν οι αλληλεπιδράσεις, απαντά σε ερωτήσεις σχετικές με τις αλληλεπιδράσεις και προτείνει “διορθωτική δράση” σε περίπτωση αρνητικών αποτελεσμάτων. Η (knowledge base) βάση γνώσης των τύπων των ναρκωτικών οργανώνεται σαν hierarchy of frames που συμπεριλαμβάνει πληροφορίες που αναφέρονται to specific drugs όπως φάρμακα που έχουν χημική συγγένεια με το ναρκωτικό, storage sites και χαρακτηριστικά αλληλεπίδρασης. Η γνώση σχετικά με το μηχανισμό

αλληλεπίδρασης του ναρκωτικού οργανώνεται με frames, σύμφωνα με έναν από τους 4 τύπους μηχανισμού: χημικοφυσικό, φαρμακοδυναμικό, φαρμακοκινητικό και φυσιολογικό. Το σύστημα παρέχει μια περιορισμένη φυσική γλώσσα interplace, συμπεριλαμβανομένου ενός διορθωτή συλλαβισμού αφού τα ονόματα των ναρκωτικών συνήθως δακτυλογραφούνται εσφαλμένα (ανακριβή). Το σύστημα εκτελείται σε PROLOG. Αναπτύχθηκε at Virginia Polytechnic Institute & State University κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το EEG ANALYSIS SYSTEM αναλύει ηλεκτροεγκεφαλογραφήματα (EEGS) καταγραμμένα από renal patients (ασθενείς). Το σύστημα αναλύει τα EEGS που δίνονται σ' αυτό απ' ευθείας από μια μηχανή EEG χρησιμοποιώντας μια γρήγορη Fourier transformation μέθοδο. Αυτό μετά χρησιμοποιεί τα αποτελεσματικά φασματικά χαρακτηριστικά για να ταξινομήσει το EEGS είτε σαν ομαλό, ή σαν μη ομαλό. Το σύστημα χρησιμοποιεί γνώσεις που παρέχονται από έναν professional ηλεκτροεγκεφαλογράφο και αναπαρίστανται με την μορφή κανόνων (rules). Οι κανόνες συνοδεύονται με παράγοντες βεβαιότητας (CF). Το σύστημα έχει την ικανότητα να εξηγεί τα αποτελέσματά του. Ένα rule-editing πρόγραμμα γραμμένο στην C, βοηθά τους αναπτυχθείς του συστήματος maintain και update the rules. Το σύστημα εκτελείται σε assembly και C και ενσωματώνεται σε ένα Motorola MC6801 single-chip microprocessor. Το EEG ANALYSIS SYSTEM αναπτύχθηκε στο Πανεπιστήμιο Vanderbilt κι έφθασε το στάδιο της έρευνας πρωτοτύπου.

Το EMERGE βοηθά τους γιατρούς στην ανάλυση του πόνου στο στήθος σ' ένα emergency-room environment. Το σύστημα αποφασίζει εάν ένας emergency-room ασθενής που υποφέρει από πόνο στο στήθος θα πρέπει να εισαχθεί στο νοσοκομείο. Επίσης παρέχει συμβουλές για πιθανές θεραπείες μαζί με μια ένδειξη κρισιμότητας της κατάστασης. Το EMERGE περιέχει

πραγματογνωμοσύνη παραγόμενη από υπαρκτές ιατρικές περιγραφές, γνωστές, σαν *criteria maps*, που αντανakλούν μια συλλογή γνώσεων η οποία αποκτιέται με την βοήθεια χρονίων επαφών και συμβουλίων με εμπειρογνώμονες. Το σύστημα αναπαριστά αυτές τις γνώσεις σαν κανόνες (*rules*) οργανωμένους in a hierarchy για να βελτιώσουν την παράσταση. Επίσης χειρίζεται υπολογισμούς βεβαιότητας ( *certainty calculations*) και παρέχει εξηγήσεις που δείχνουν τα λογικά *paths* που παίρνονται από *rule searches*. Το EMERGE εκτελείται σε PASCAL και λειτουργεί on mainframes minicomputers και microcomputers. Το σύστημα αναπτύχθηκε στην UCLA κι έφθασε το στάδιο field πρωτοτύπου.

Το EXAMINER αναλύει των γιατρών την διαγνωστική συμπεριφορά σε υποθέσεις εσωτερικής ιατρικής (*internal medicine*). Το σύστημα παρουσιάζει μια υποθετική περίπτωση και ο γιατρός αναγνωρίζει την ασθένεια και τα συμπτώματα τα οποία την ακολουθούν και στη συνέχεια αναφέρεται σε άλλες ασθένειες η προβλήματα που θα μπορούσαν να παρουσιασθούν. Το σύστημα εκτελεί αυτή την ανάλυση χρησιμοποιώντας: 1) ευριστικές γνώσεις για το πως συσχετίζονται ασθένειες με τα συμπτώματά τους και 2) ιατρικές γνώσεις που κατέχονται απ' ευθείας από the data base INTERNIST. Το σύστημα παράγει ένα κείμενο σχολίων που αξιολογεί του γιατρού την διάγνωση, αναφέροντας τις σωστές υποθέσεις κι εξηγώντας που οι εσφαλμένες είναι λάθος. Η γνώση είναι στην μορφή των *procedures* που ενσωματώνουν αρχές εσωτερικής ιατρικής. Το σύστημα εκτελείται σε LISP και αναπτύχθηκε στο Πανεπιστήμιο του Pittsburgh. Έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το GALEN κάνει διάγνωση σε περιπτώσεις παιδιών που έχουν εκ γενετής καρδιακές παθήσεις. Το σύστημα χρησιμοποιεί δεδομένα που περιγράφουν το ιατρικό ιστορικό του ασθενούς, τις φυσικές εξετάσεις, ακτίνες-X και EKGs για να αναγνωρίζουν την ασθένεια που παρουσιάζεται. Το GALEN κάνει υποθέσεις για ένα μικρό σύνολο πιθανών ασθενειών και τις

βάσει σε προτεραιότητα. Η σειρά προτεραιότητας βασίζεται στο βαθμό που ταιριάζουν τα αναμενόμενα της κάθε ασθένειας με τις πραγματικές data values του ασθενούς. Του GALEN η πραγματογνωμοσύνη ( expertise) βασίζεται σε μοντέλα παιδιατρικών καρδιολογικών ασθενειών. Οι γνώσεις αντιπροσωπεύονται σαν συνδυασμός κανόνων (rules) και πλαισίων (frames). Με τους κανόνες περιγράφονται οι συνθήκες κάτω από τις οποίες μια υπόθεση θα έπρεπε να θεωρείται, να δέχεται, να απορρίπτεται ή να τροποποιείται. Το GALEN χρησιμοποιεί τα frames για να συλλέγει πληροφορίες σχετικές to a particular disease hypothesis. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Minnesota κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το GUIDON δίνει οδηγίες στους μαθητευόμενους γιατρούς για τη συλλογή αντιμικροβιακής θεραπείας σε νοσοκομειακούς ασθενείς με βακτηριακές μολύνσεις. Το σύστημα συλλέγει μια περίπτωση και αφού την λύσει, την παρουσιάζει στο φοιτητή για να τη λύσει και αυτός και ύστερα αναλύει τις απαντήσεις και τα ερωτήματα του φοιτητή που εμφανίζονται κατά το solution process. Κατόπιν, με βάση αυτήν την διαδικασία το σύστημα καθορίζει πόσο στενά οι γνώσεις και ο συλλογισμός του φοιτητή ταιριάζουν με την πορεία της διάγνωσης (diagnosis procedure) που το σύστημα χρησιμοποίησε για να λύσει την περίπτωση. Οι διαφορές που βρίσκονται αποτελούν έναν οδηγό με την βοήθεια του οποίου γίνεται η παρουσίαση των explanation mechanism του συστήματος. Το GUIDON στην πραγματικότητα χρησιμοποιεί το MYCIN expert σύστημα για να λύσει τις περιπτώσεις που του παρουσιάζονται. Στην ουσία διδάσκει στους φοιτητές τους κανόνες (rules) και τις procedures που ενσωματώνονται στο MYCIN. Το GUIDON είναι rule-based σύστημα που περιέχει και τα δύο : rules και metarules ( κανόνες που αποφασίζουν πως να χρησιμοποιούν τους κανόνες). Αναπτύχθηκε σε INTERLISP και εφαρμόσθηκε στο Πανεπιστήμιο Stanford. Το σύστημα έφθασε το στάδιο έρευνας πρωτοτύπου.

Το HDDSS βοηθά τους γιατρούς να καθορίσουν και να επιλέξουν κατάλληλη θεραπεία για ασθενείς που πάσχουν από την νόσο του Hodgkin. Το σύστημα εφαρμόζει Bayesian estimation techniques για να παράγει a priori probabilities της έκτασης που έχει καταλάβει ο όγκος. Μετά χρησιμοποιεί τις πιθανότητες αυτές για να επιλέξει μιά diagnostic procedure χρησιμοποιεί το θεώρημα του Bayes για να επαναθεωρήσει τις πιθανότητες και να συνεχέσει the process. Το HDDSS τελικά παράγει ένα optimal diagnostic plan και ένα σετ των πιο κοντινών optimal plans τα οποία μπορεί να μελετήσει ο γιατρός. Το σύστημα περιέχει γνώσεις ραδιοθεραπείας σε μια ιεραρχική ταξινόμηση και patient data σε μιά relational data base. Το HDDSS εκτελείται σε MACLISP. Αναπτύχθηκε στο MIT κι έφθασε το στάδιο μιάς έρευνας πρωτοτύπου.

Το HEADMED συμβουλεύει τους γιατρούς σε θέματα κλινικής ψυχοφαρμακολογίας. Κάνει διάγωση σε μιά σειρά ψυχιατρικών “διαταραχών” και συστήνει φαρμακευτική αγωγή. Το σύστημα σχεδιάστηκε για να βοηθά στην διδασκαλία και να παρέχει συμβουλές. Το HEADMED χρησιμοποιεί προσεγγιστικά 120 κλινικές παραμέτρους για να εκτελεί τη διαγνωσή του και την σύσταση της θεραπείας (π.χ. δόση φαρμάκου και διάρκεια θεραπείας). Επίσης περιέχει γνώσεις που αφορούν the differential diagnosis απο τις κυριότερες επηρεαστικές διαταραχές όπως την σχιζοφρένεια και την γενική κατηγορία οργανικών εγκεφαλικών διαταραχών. Η βάση γνώσης (knowledge base) επίσης συμπεριλαμβάνει τον (MMPI) τον Minnesota Multiphasic Personality Inventory καθώς και skeletal knowledge που αφορά νευρώσεις και διαταραχές της συμπεριφοράς. Το HEADMED γράφεται σε EMYCIN ( την εξήγηση, την απόκτηση των γνώσεων και άλλων ευκολιών υποστήριξης). Αναπτύχθηκε στο University of California Irvine κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το HEART IMAGE INTERPRETER βοηθά τους γιατρούς να εκτελούν για διαγνωστική ερμηνεία της κινητικής συμπεριφοράς της καρδιάς. Το σύστημα αναλύει δυσδιάστατες εικόνες κατανομής έντασης της καρδιάς. Αυτές δημιουργούνται από την παρουσία του Technetium 99-m (radionuclide υλικό) που βρίσκεται στον οργανισμό μετά από ενδοφλέβια ένεση που γίνεται στον ασθενή. Μια σπινθηροβόλουσα μηχανή, παράγει μια σειρά από έως 64 εικόνες για το σύστημα (structural) ιδιότητες της καρδιάς και του αριστερού κόλπου της καρδιάς, τις κινητικές φάσεις ενός καρδιακού κύκλου και τους κανόνες για σχετική ιατρική απόδειξη προς διάγνωση. Η εικόνα της πορείας και ιατρικές διαγνωστικές γνώσεις (image processing, medical diagnostic knowlegde) αναπαρίστανται με την βοήθεια ενός semantic net που περιέχει frames και σχετικούς κανόνες που χρησιμοποιούν παράγοντες βεβαιότητας. Το σύστημα εκτελείται στην RATFOR, μια διάλεκτο της FORTRAN. Αναπτύχθηκε στο Πανεπιστήμιο του Erlangen κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το HEME βοηθά τους γιατρούς να κάνουν διάγνωση αιματολογικών ασθενειών. Το σύστημα χρησιμοποιεί patient findings που εισέρχονται από έναν γιατρό και μια εκδοχή του θεωρήματος του Bayes για να υπολογίζει την πιθανότητα σύμφωνα με την οποία ο ασθενής έχει κάποια από τις ασθένειες που καταγράφονται στο σύστημα. Το HEME μετά επιδεικνύει έναν κατάλογο διαφορετικών διαγνώσεων μαζί με την πιθανότητα που έχουν για να συμβούν. Η γνώση στο HEME συμπεριλαμβάνει εκτιμήσεις της συχνότητας του συμβάντος μιας δεδομένης ασθένειας, την πιθανότητα ότι ο ασθενής με την ασθένεια έχει ένα given findings καθώς και την πιθανότητα ο ασθενής χωρίς την ασθένεια να έχει given finding. Αυτές οι conditiongal πιθανότητες αντιπροσωπεύουν τις κρίσεις των εμπειρογνομόνων κλινικών αιματολογίας και συμπεριλαμβάνουν μέτρα του πόσο έμπιστοι είναι οι κλινικοί σχετικά με τις original εκτιμήσεις πιθανότητας. Το HEME αναπτύχθηκε στο Πανεπιστήμιο Cornell κι έφθασε το στάδιο field πρωτοτύπου.

Το HT-ATTENDING κριτικάρει μια προσέγγιση γιατρού στην “φαρμακολογική διαχείριση” βασικής υπέρτασης. Το σύστημα βοηθά τους γιατρούς να κουράρουν υπέρτασικούς ασθενείς και παρέχει πληροφορίες για καινούργια φάρμακα και θεραπείες. Επεκτείνει “την Προσέγγιση” που χρησιμοποιείται στο ATTENDING to the domain της υπέρτασης. Το σύστημα χρησιμοποιεί πληροφορίες για τον ασθενή (π.χ. ηλικία, ιατρικά προβλήματα) μαζί με πληροφορίες για την επι του παρόντος αντιυπερτασική δίαιτα του ασθενούς και τις προτιθέμενες αλλαγές σ’αυτήν. Επίσης κριτικάρει τις προταθείσες αλλαγές. Η κριτική δίνεται υπο μορφή παραγράφων του Αγγλικού κειμένου. Το σύστημα περιέχει γνώσεις για αντιυπερτασικούς παράγοντες (για αντιυπερτασικά μέσα) που χρησιμοποιούνται για θεραπείες ενός ασθενούς, τυπικότητες θεραπείας και συνθήκες που επηρεάζουν την κούρα της υπέρτασης. Το σχήμα αναπαράστασης γνώσης που χρησιμοποιείται απο το συστημα είναι frame-base. Το σύστημα υλοποιήθηκε με LISP. Αναπτύχθηκε στο Παμεπιστήμιο Yale κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το INTERNIST-I/CADUCEOYS βοηθά τον γιατρό στο να κάνει πολλαπλές και πολύπλοκες διαγνώσεις στην γενική εσωτερική ιατρική δίνοντας του το ιστορικό του ασθενούς και τα συμπτώματα ή αποτελέσματα εργαστηριακών τέστ. Το σύστημα βασίζει τις αποφάσεις του σ’ένα σύνολο ασθενειών που περιέχουν findings που συμβαίνουν σε σχέση με κάθε ασθένεια. Το σύστημα είναι ένα απο τα πιο μεγάλα ιατρικά expert συστήματα που αναπτύχθηκαν περιέχοντας φακέλους που αναφέρονται σε πάνω απο 500 ασθένειες και περιγράφουν περισσότερες απο 3.500 εκδηλώσεις της ασθένειας. Οι γνώσεις στο σύστημα αντιπροσωπεύονται σαν network of findings και ασθενειών και is accesed based στους περιορισμούς της ταξινομίας της ασθένειας και σχέσεων αιτιολογίας. Το σύστημα εκτελείται σε LISP. Η πρώτη εκδοχή του συτήματος ονομάσθηκε



INTERNIST-I η δεύτερη CADUCEOUS. Αναπτύχθηκε στο Πανεπιστήμιο του Pittsburgh κι έφθασε το στάδιο field πρωτοτύπου.

Το IRIS βοηθά τους γιατρούς να κάνουν διάγνωση και θεραπεία ασθενειών. Το σύστημα συλλέγει πληροφορίες απο τον γιατρό σχετικά με τα συμπτώματα ασθενούς, αναπτύσσοντας έτσι ένα σύνολο πιθανών διαγνώσεων. Το IRIS διαλέγει μια θεραπεία που ανταποκρίνεται στην διάγνωση. Η ιατρική διαγνωστική πραγματογνωμοσύνη αντιπροσωπεύεται, χρησιμοποιώντας ένα σημαντικό δίκτυο με την βοήθεια του οποίου προσδιορίζονται οι σχέσεις ανάμεσα σε συμπτώματα ασθένειες και θεραπείες. Η πορεία συμπεράσματος(inference process) ελέγχεται με decision tables που συνδεονται με τους κόμβους του σημαντικού δικτύου. Το IRIS προπλάσσει MYCIN-LIKE παράγοντες βεβαιότητας διαμέσου του δικτύου για να το βοηθήσει στο να διαλέγει μια διάγνωση. Το IRIS αναπτύχθηκε στο Πανεπιστήμιο Rutgers και είναι ακόμη ενα περιβάλλον για εξερεύνηση ιατρικής απόφασης που παίρνεται, παρά ενα πραγματικό expert-system.

Το MDX κάνει διάγνωση της ύπαρξης και αιτίας του συνδρόμου του συκωτιού, γνωστό σαν cholestasis . Το σύστημα βασίζει την διάγνωση του σε ιστορικό ασθενούς, σημεία, συμπτώματα, και clinical data. Το MDX λειτουργεί σαν κοινότητα συνεργασίας expert diagnosticians, καθένας απο τους οποίους έχει διαφορετικές ειδικότητες. Αυτοί οι εμπειρογνώμονες καλουν ο ένας τον άλλον για να επιλύουν τα προβλήματα που απαιτούν ειδικές γνώσεις και πραγματογνωμοσυνη. Η επικοινωνία γίνεται διαμέσου black-board μηχανισμού. Η πραγματογνωμοσύνη του MDX αποτελείται απο diagnostic heuristics και ενα ιεραρχικό βαθύ μοντέλο της conceptual structure της cholestasis. Το PATREC and RADEX είναι μέλη της “κοινότητας” των expert systems που καλούνται απο το MDX. Το MDX εκτελείται σεLISP. Αναπτύχθηκε στο Ohio State University κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το MECS-AI βοηθά τους γιατρούς να κάνουν διαγνώσεις και να προτείνουν θεραπείες για καρδιαγγειακές και θυροειδούς ασθένειες. Αν το αναπτυγμένο ειδικά για καρδιαγγειακές ασθένειες, αυτό διορθώθηκε σε ένα πιο γενικό εργαλείο για ανάπτυξη ιατρικών συστημάτων συμβούλευσης και εφαρμόστηκε σε ασθένειες θυροειδούς. Το MECS-AI χρησιμοποιεί σαν γνώσεις για καρδιαγγειακές παθήσεις και ανωμαλίες θυροειδούς τις διαγνώσεις που του παρέχονται από τους ιατρικού εμπειρογνώμονες. Είναι ένα backward chaining, rule based system και περιέχει έναν knowledge-base editor για να ευκολύνει defining και modifying system expertise. Δυο εκδοχές του MECS-AI υπάρχουν, μια που εκτελείται στην INTERLISP και η άλλη στην EPICS-LISP. Αναπτύχθηκε στο University of Tokyo Hospital κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το MEDICO δίνει συμβουλές στους οφθαλμολόγους για την διάγνωση και θεραπεία ασθενειών του αμφιβληστροειδούς. Το σύστημα περιέχει γενικές κλινικές γνώσεις και μια μεγάλη βάση δεδομένων των γεγονότων σχετικά με προηγούμενους ασθενείς και γεγονότα. Το MEDICO είναι ένα forward chaining, rule based σύστημα που χειρίζεται certainty συσχετίζοντας εκτιμήσεις πιθανοτήτων που παρέχονται από κύριους εμπειρογνώμονες με τους κανόνες (rules). Ο interactive rule acquisition από τους κύριους εμπειρογνώμονες (domain experts), υποστηρίζεται από ένα knowledge acquisition and maintenance module (KAMM). Ένα άλλο module, το RAIN (Relational Algebraic Interpreter). Σχετικός Αλγεβρικός Μεταφραστής υποστηρίζει την εξέταση, την χρονολογία ως το παρόν (update) και την ξαναοργάνωση των κανόνων του knowledge base. Το MEDICO εκτελείται στην C. Αναπτύχθηκε στο Πανεπιστήμιο του Illinois κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το MED1 βοηθά τους γιατρούς να κάνουν διάγνωση σε ασθένειες που σχετίζονται με πόνο στο στήθος. Το σύστημα χρησιμοποιεί findings όπως συχνότητα καρδιακών χτύπων και πίεση αίματος τα οποία βοηθούν το σύστημα να καθορίσει την ασθένεια ή τις ασθένειες. Οι υποθέσεις ασθενειών προτείνονται από forward-chaining διαμέσου των κανόνων που περιέχουν προτεινόμενες αποδείξεις. Η top-ranking διάγνωση υπόκειται κατόπιν σε περαιτέρω αξιολόγηση με την βοήθεια backward chaining μέσω MYCIN-LIKE rules που είτε επιβεβαιώνουν ή προτείνουν εναλλακτικές λύσεις. Το MED1 χρησιμοποιεί μια τροποποιημένη version of MYCIN's certainty factor scheme, στο οποίο τα αποτελέσματα μιας διάγνωσης αρρώστιας μετασχηματίζεται σε μια από τις επτά probability classes από "excluded" σε "confirmed". Το σύστημα επίσης παρέχει απόκτηση γνώσης και explanation facilities (επεξηγηματικές ευκολίες). Το MED1 εκτελείται στην INTERLISP. Αναπτύχθηκε στο Πανεπιστήμιο του Kaiserslautern κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το MI βοηθά τους γιατρούς να κάνουν διάγνωση μυοκαρδίου εμφράγματος (infraction) διαμέσου ανάλυσης της δραστηριότητας ενζύμων. Το σύστημα φθάνει στην διάγνωση της καρδιακής πάθησης, ελέγχοντας τα ανυψωμένα επίπεδα ορισμένων ενζύμων στο αίμα, πάνω από περίοδο αρκετών ημερών. Του MI η πραγματογνωμοσύνη συμπεριλαμβάνει γνώσεις στο πως να χειρίζεται κάποιος the time- dependant nature of the medical findings, συμπεριλαμβανομένων και τεχνικών για αυτόματη χρονολόγηση (updating), διόρθωση (revising) και interpreting a patient's record. Το MI είναι ένα forward-chaining, rule-based σύστημα. Εκτελείται σε μια version of EXPERT τροποποιημένο για να παρουσιάσει και να χειριστεί time-dependent rules και data. Το MI αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το MODIS βοηθά τους γιατρούς να κάνουν διάγνωση σε διάφορους τύπος της αρτηριακής υπέρτασης. Το σύστημα πρώτα συλλέγει πληροφορίες για τους ασθενείς όπως συμπτώματα και αποτελέσματα εργαστηριακών αποτελεσμάτων. Κατόπιν κάνει υποθέσεις για ομάδες ασθενειών που έχουν χαρακτηριστικά συνεπή με τις συλλεχθείσες πληροφορίες. Εάν υπάρχουν διάφορες πιθανότητες το σύστημα συλλέγει εκείνη την ομάδα των ασθενειών που ranked most likely από τους εμπειρογνώμονες γιατρούς. Το σύστημα μπορεί εξηγήσει την λογικότητα του κάνοντας ερωτήσεις, σχηματίζοντας διαγνώσεις και εξετάζοντας υποθέσεις. Η διαγνωστική πραγματογνωμοσύνη που ενσωματώνεται στο MODIS προέρχεται από χειρούργους και θεραπευτικούς εμπειρογνώμονες στον κλάδο. Αυτή η γνώση αντιπροσωπεύεται σαν ένα σημαντικό δίκτυο που περιέχει πλαίσια (frames) των οποίων τα slots περιέχουν κανόνες, καθώς επίσης και data. Το σύστημα εκτελείται σε LISP. Αναπτύχθηκε στο Tbilisi, στη Γεωργιανή Ρωσία κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το MYCIN βοηθά του γιατρούς στην επιλογή κατάλληλης αντιμικροβιακής θεραπείας για ασθενείς νοσοκομείου με βακτηρεμία, μηνιγγίτιδα, και μολύνσεις κυστίτιδας. Το σύστημα κάνει διάγνωση της αιτίας της μόλυνσης χρησιμοποιώντας γνώσεις σχετικά με την μόλυνση οργανισμών, με το ιστορικό του ασθενούς, τα συμπτώματα και τα αποτελέσματα εργαστηριακών εξετάσεων. Το σύστημα συστήνει θεραπεία φαρμάκων (τύπο και δόση) σύμφωνα με procedures που ακολουθούνται από γιατρούς έμπειρους σε θεραπείες μολυσματικών ασθενειών. Το MYCIN είναι ένα rule-based σύστημα με σχήμα ελέγχου backward-chaining. Συμπεριλαμβάνει μηχανισμούς για εκτέλεση υπολογισμών βεβαιότητας και για παροχή εξηγήσεων της λογικής πορείας του συστήματος. Το MYCIN εκτελείται σε LISP. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το NEOMYCIN βοηθά τους γιατρούς να κάνουν διάγνωση και να θεραπεύσουν ασθενείς με μηνιγγίτιδα και παρόμοιες ασθένειες. Το σύστημα συσσωματώνει πραγματογνωμοσύνη παραγόμενη από το MYCIN, αντιπροσωπούμενη με έναν τρόπο που διευκολύνει εξήγηση και διδασκαλία. Η γνώση του συστήματος είναι σε τύπο κανόνων (rules) που οργανώνονται by a disease hierarchy και ελέγχονται από forward-chaining metarules που αποτελούν την διαγνωστική procedure. Η διαφορά κλειδί ανάμεσα στο MYCIN και NEOMYCIN είναι ο σαφής χωρισμός της διαγνωστικής procedure από την γνώση της ασθένειας. Το NEOMYCIN παρέχει εξηγήσεις της διαγνωστικής στρατηγικής του. Το GUIDON 2 χρησιμοποιεί τη βάση γνώσης του NEOMYCIN και τους διαγνωστικούς του metarules καθώς και την πηγή του των υλικών διδασκαλίας. Το NEOMYCIN εκτελείται στην INTERLISP-D. Αναπτύχθηκε στο Stranford κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το NEUREX βοηθά τους γιατρούς να κάνουν διάγνωση σε ασθενείς με αρρώστιες του νευρικού συστήματος. Το σύστημα χρησιμοποιεί τα αποτελέσματα νευρολογικής εξέτασης που γίνεται σε λιπόθυμους (unconscious) ασθενείς για να χαρακτηρίσει την ζημιά του νευρικού συστήματος και να ταξινομήσει τους ασθενείς σύμφωνα με τον τύπο της ζημιάς. Το NEUREX είναι ένα rule-based σύστημα που απασχολεί και τα δύο forward και backward chaining καθώς επίσης και ένα MYCIN-like certainty factor mechanism. Κανόνες που περιέχουν νευρολογική εντόπιση πραγματογνωμοσύνης οργανώνονται σε μια ιεραρχία συμπεράσματος στην οποία διαδοχικά υψηλότερα επίπεδα αφαίρεσης πληροφοριών σχετικά με τον ασθενή. Το NEUREX εκτελείται σε Wisconsin LISP. Αναπτύχθηκε στο Πανεπιστήμιο της Maryland κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το NEUROLOGISTI-I βοηθά τους γιατρούς να κάνουν διάγνωση νευρολογικών διαταραχών εντοπίζοντας οργανικές βλάβες που συμβαίνουν

μέσα στο κεντρικό νευρικό σύστημα (CNS). Το σύστημα αναλύει δεδομένα ασθενειών συμπεριλαμβανομένων νευρολογικών παραγόντων και αποτελέσματα φυσικής εξέτασης και μετά παράγει μια περίληψη κακής λειτουργίας χωρών (tracts) “χαρτογραφώντας” συμπτώματα το tract status. Το σύστημα χρησιμοποιεί φυσιολογικές γνώσεις (physiological) που ενδοκωδικοποιούνται σαν ένα αναλογικό γεωμετρικό μοντέλο του CNS στο οποίο τα nervous tract cross-sections προσεγγίζονται by polygons που αναπαρίστανται σαν σύνολα κορυφών. Το NEUROLOGIST-I εκτελείται στην FRANZLISP. Αναπτύχθηκε στο State University of New York at Buffalo κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το OCULAR HERPES MODEL βοηθά έναν γιατρό να κάνει διάγνωση και να θεραπεύσει ασθενείς με πολύπλοκο οφθαλμικό έρπη. Το σύστημα συνδέει το ιστορικό κλινικού ασθενούς και τα ευρήματα του εργαστηρίου με κατηγορίες ασθενειών και χρησιμοποιεί αυτές τις γνώσεις για να κάνει διάγνωσης της ασθένειας (π.χ. ο ασθενείς έχει μια οργανική βλάβη επιθηλιακού κερατοειδούς χιτώνα) και συστήνει θεραπεία (π.χ. δίδει την αλοιφή vidarabine 5 φορές ημερησίως). Το σύστημα συλλέγει θεραπεία βασισμένη σε αποτελεσματικότητα φαρμάκων και στην αντίσταση του ασθενούς στο φάρμακο και στην αλλεργική αντίδραση. Το σύστημα είναι ένα forward chaining, rule-based σύστημα που εκτελείται σε EXPERT. Αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε το στάδιο πρωτότυπης επίδειξης.

Το ONCOCIN βοηθά τους γιατρούς στο να ασχολούνται με την θεραπεία ασθενών που πάσχουν από καρκίνο και που υποβάλλονται σε πειράματα χημειοθεραπείας τα οποία ονομάζονται protocols. Το σύστημα συλλέγει θεραπεία συσχετίζοντας πληροφορίες για την διάγνωση του ασθενή, προηγούμενες θεραπείες κι εργαστηριακά τεστ με την knowledge about protocols-past πειράματα. Το σύστημα περιέχει γνώσεις για 34 νόσους τους

Hodgkin και lymphoma protocols. Το ONCOCIN είναι ένα rule-based σύστημα που χρησιμοποιεί και τα δύο: forward και backward chaining. Εκτελείται σε INTERLISP. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το PATHFINDER βοηθά τους παθολόγους να ερμηνεύσουν findings που προέρχονται από την μικροσκοπική εξέταση of lymph node tissue. Το σύστημα θέτει ερωτήσεις στον παθολόγο με έναν τρόπο σχεδιασμένο για να μειώσει την αβεβαιότητα στη differential diagnosis και μπορεί να παρέχει μια δικαιολογία για τη πιο πρόσφατη ερώτηση που ετέθηκε. Το σύστημα βασίζει τις αποφάσεις του σε ένα σετ φακέλων ασθενειών που περιέχουν findings τα οποία φυσιολογικά συσχετίζονται με κάθε ασθένεια. Αυτή η πραγματογνωμοσύνη στην διάγνωση του παθολογικού κόμβου της δαμάλιδος (lymph) προέρχεται από εμπειρογνώμονες αιματοπαθολόγους. Η γνώση του PATHFINDER ενδοκωδικοποιείται χρησιμοποιώντας frames. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε στο στάδιο μιας πρωτότυπης επίδειξης.

Το PATREC διευθύνει μια βάση δεδομένων (data base) of patient (ασθενής) records ασθενούς, παρέχοντας γιατρούς κατάλληλους για διάγνωση και το MDX σύστημα εμπειρογνώμωνων το οποίο έχει access στα records. Χειρίζεται δεδομένα ασθενούς στη συνάφεια της διάγνωσης για το σύνδρομο που ονομάζεται cholestasis. Το PATREC δέχεται δεδομένα από τον χρήστη, τα αποθηκεύει κατάλληλα, παρέχει μια γλώσσα έρευνας για το ζήτημα που απαντά, προετοιμάζει περιληπτικές αναφορές και κάνει προτάσεις που βοηθούν στην διάγνωση. Η γνώση στο PATREC συμπεριλαμβάνει ένα conceptual model ιατρικών δεδομένων, (π.χ. η σπουδαιότητα ενός ειδικού εργαστηριακού τεστ και οι αναμενόμενες του αξίες) καθώς και ένα μοντέλο ασθενή (π.χ. προσωπικά ιστορικά ασθενούς και κλινικά επεισόδια). Η γνώση αντιπροσωπεύεται με την μορφή frames με συναπτόμενους κανόνες οι οποίοι

χρησιμεύουν για να εποπτεύουν bookkeeping functions data input και automatic temporal inferencing. Το PATREC εκτελείται στην LISP. Αναπτύχθηκε στο Πανεπιστήμιο του Οχάιο κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το PEC βοηθά τους πρώτους εργατές της υγείας να κάνουν διάγνωση και να θεραπεύσουν τις “ανωμαλίες” που προκαλούν τύφλωση. Το αρχικό του σύνολο των ναι και όχι ερωτήσεων (όπως είναι το μάτι κόκκινο) οδηγεί σε περαιτέρω σύνολο ερωτήσεων σχετικές με τον ασθενή. Όταν έχουν συγκεντρωθεί αρκετές πληροφορίες το σύστημα παράγει περίληψη της υπόθεσης, ακολουθούμενο από διαγνωστικά του συμπεράσματα και συστάσεις management. Του PEC οι γνώσεις προέρχονται από τον Οργανισμό Παγκόσμιας Υγείας που οδηγούν σε πρωταρχική φροντίδα οφθαλμού και αντιπροσωπεύεται υπό μορφή rules χρησιμοποιώντας ένα forward chaining inference scheme. Το σύστημα αρχικά εκτελέσθηκε στην EXPERT, μετά μεταφράσθηκε αυτόματα στην BASIC για χρήση των μικροκομπιούτερ. Το PEC αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε το στάδιο μιας πρωτότυπης έρευνας.

Το PIP βοηθά τους γιατρούς παίρνοντας το ιστορικό της παρούσας ασθένειας ενός ασθενή με οίδημα. Το σύστημα συνυφαίνει the process of information gathering and diagnosis, εναλλάσσοντας ανάμεσα στις ερωτήσεις που γίνονται για την απόκτηση καινούργιας πληροφορίας και ολοκληρώνοντας αυτές τις καινούργιες πληροφορίες σε μια developing picture του ασθενούς. Οι γνώσεις που περιέχονται στο PIP συμπεριλαμβάνουν πρωτοτυπικά findings όπως: signs, συμπτώματα δεδομένα εργαστηρίων, τον χρόνο πορείας της δεδομένης ασθένειας και κανόνες για κρίση του πόσο στενά ένας δεδομένος ασθενής ταιριάζει με μια υποτιθέμενη αρρώστια ή κατάσταση. Οι ερωτήσεις του PIP ελέγχονται από ένα σεντ διαγνωστικών υποθέσεων. Η ικανότητα των υποθέσεων να λογαριάσουν τα findings της περίπτωσης υπολογίζεται κατά μέσον όρο the measure of fit to arrive at a final certainty measure for ranking



hypothesis. Η γνώση στο PIP αντιπροσωπεύεται χρησιμοποιώντας frames. Το PIP εκτελείται σε CONNIVER κι αναπτύχθηκε στο MIT. Έφθασε το στάδιο μιας πρωτότυπης έρευνας.

Το PUFF κάνει διάγνωση στην παρουσία και σοβαρότητα της ασθένειας πνευμόνων σ'έναν ασθενή, ερμηνεύοντας μετρήσεις από αναπνευστικά τεστ που διαχειρίζονται σε εργαστήριο πνευμονικής λειτουργίας. Τα δεδομένα που ερμηνεύονται συμπεριλαμβάνουν αποτελέσματα τεστ (π.χ. ολική των πνευμόνων χωρητικότητα υπόλοιπο όγκου) και ιστορικά ασθενούς. Το σύστημα βασιζέται τις αποφάσεις του σε γνώση σχετικά με τα είδη των αποτελεσμάτων των τεστ που παράγονται από διαφορετικές πνευμονικές ανωμαλίες. Το PUFF είναι ένα backward-chaining, rule-based σύστημα που εκτελείται σε EMYCIN. Αναπτύχθηκε στο Πανεπιστήμιο Stanford και δοκιμάστηκε στο Ειρηνικό Ιατρικό Κέντρο του San Francisco. Έφθασε το στάδιο πρωτότυπης παραγωγής.

Το RADEX βοηθά τους φυσικούς και το MDX σύστημα εμπειρογνομόνων να κάνουν διάγνωση για την cholestasis (σύνδρομο συκωτιού). Αυτός ο σύμβουλος της ραδιολογίας κάνει περιληπτικές περιγραφές ιατρικών εικόνων ασθενούς και τις χρησιμοποιεί σαν βάση για απάντηση ερωτήσεων σχετικά με ανωμαλίες ανατομικές ή φυσιολογικές του ασθενούς. Του συστήματος η πραγματογνωμοσύνη αποτελείται από ένα μοντέλο conceptual συναφών οργάνων και οργανικών ανωμαλιών που μπορούν να εμφανισθούν σε διάφορες εικόνες. Η γνώση αναπαρίσταται με frames που περιέχουν περιγραφές, default values, relationship και συναπτόμενες proceduses. Αυτές είναι 4 ειδών: 1)The imaging procedures (μερικές από αυτές αναφέρονται σε λεπτομερείς εξετάσεις συκωτιού και σε περιγραφές υπερηχογραφημάτων) 2) major organs procedures (αφορούν: κύστη του συκωτιού και χοληδόχο κύστη) 3)parts of organs procedures (αφορούν : τον λοβό ) και 4) obuormalities and deformities in the organs

procedures (αφορούν : πρήξιμο (όγκο) και στένωμα). Το RADEX εκτελείται στην LISP. Αναπτύχθηκε στο Ohio State University κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το RX βοηθά τους χρήστες να εκτελούν μελέτες σε μεγάλες μη τυχαίες, προσανατολισμένες χρονικά clinical data bases αυτοματοποιώντας την process of hypothesis generation καθώς και την επεξηγηματική ανάλυση (exploratory analysis). Το RX έχει εφαρμοσθεί στο πρόβλημα του να βρεις αιτιολογικές σχέσεις (causal relationships) within the American Rheumatism Association Medical Information System (ARAMIS) data base. Το σύστημα χρησιμοποιεί μη παραμετρικές συσχετίσεις για να παράγει καταλόγους υποτιθέμενων σχέσεων. ΟΙ υποθέσεις δοκιμάζονται χρησιμοποιώντας κατάλληλες στατιστικές μεθόδους και τα θετικά αποτελέσματα συσσωματώνονται στη βάση δεδομένων. Το RX περιέχει μια βάση γνώσεων tree-structured η οποία αντιπροσωπεύει μια ταξινόμηση σχετικών απόψεων ιατρικής. Το RX εκτελείται σε INTERLISP. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο πρωτότυπης έρευνας.

Το SPE κάνει τις διακρίσεις ανάμεσα στις διάφορες αιτίες των ερεθιστικών καταστάσεων ενός ασθενούς (π.χ.. κύρωση συκωτιού, μυέλωμα-έναν τύπος καρκίνου) ερμηνεύοντας τύπους κυμάτων από συσκευή που ονομάζεται scanning densitometer (πυκνόμετρο). Το σύστημα κάνει ερμηνείες, εφαρμόζοντας γνώσεις για το πως τα αναγνώσματα των οργάνων και τα δεδομένα ασθενούς αναφέρονται σε κατηγορίες ασθενών Το SPE είναι ένα forward chaining, rule-based σύστημα που πρωτοεκτελέσθηκε σε EXPERT και μετά μεταφράσθηκε σε assembly language για τον Motorola 6809 microprocessor. Το σύστημα αναπτύχθηκε στο Πανεπιστήμιο Rutgers και έχει συσσωματωθεί στο Cliniscan, ένα εξονυχιστικό πυκνόμετρο ( densitometer) αγορασμένο από τα χημικά εργαστήρια Helena Το SPE έφθασε το στάδιο ενός εμπορικού συστήματος.

Το SYSTEM D βοηθά τους γιατρούς να κάνουν διάγνωση της πιθανής αιτίας της ζάλης (του ίλιγγου) σ'έναν ασθενή. Το σύστημα παρακινεί (υπενθυμίζει) στον γιατρό να εισέλθει σε ευρήματα (π.χ. τρέχουσες θεραπείες) και εκδηλώσεις (π.χ. αίσθηση επικείμενης λιποθυμίας) και παράγει διάγνωση, συμπεριλαμβάνοντας σειρά συναγωνιστικών εναλλακτικών λύσεων. Το SYSTEM D χειρίζεται καταστάσεις που πολλαπλές αιτίες ίλιγγου μπορούν να παρουσιασθούν ταυτόχρονα. Οι γνώσεις που περιέχονται στο SYSTEM D συμπεριλαμβάνουν και τα δυο: case-specific πληροφορίες και πραγματογνωμοσύνη που αφορά τη διάγνωση των αιτιών της ζάλης. Η διαγνωστική γνώση κατανέμεται σε πολλαπλές ιατρικές ειδικότητες που ανταποκρίνονται στις αριθμούσες δυνατές αιτίες της ζάλης. Το σύστημα χρησιμοποιεί ένα frame-based σχήμα αντιπροσώπευσης γνώσης και έναν sedquential generate-and-test inference mechanism (μηχανισμός συμπεράσματος). Το SYSTEM D εκτελείται στην KMS. Αναπτύχθηκε στο Πανεπιστήμιο της Maryland κι έφθασε το στάδιο έρευνας πρωτοτύπου.

Το THYROID MODEL βοηθά τους γιατρούς να κάνουν διάγνωση στις ανωμαλίες του θυροειδούς (π.χ. υποθυροειδισμός). Το σύστημα δέχεται ένα αρχικό σετ ευρημάτων ασθενών (π.χ. δημογραφικές πληροφορίες, συμπτώματα, αποτελέσματα εργαστηριακών τεστ) και παρακινεί για τα επιπλέον δεδομένα που χρειάζονται για μια διάγνωση. Η πραγματογνωμοσύνη του συστήματος αποτελείται από διαγνωστικούς κανόνες λογικής και από μια ταξινομητική κατασκευή (a taxonomic structure) της λειτουργίας θυροειδούς και της θυροειδούς παθολογίας. Το THYROID MODEL είναι ένα forward chaining, rule-based σύστημα. Εξηγεί γιατί υποθέσεις επιβεβαιώνονται δείχνοντας τους κανόνες απόφασης που εφαρμόζονται και τους συσχετισμένους παράγοντες βεβαιότητας. Το THYROID MODEL εκτελείται στην EXPERT...Αναπτύχθηκε στο Πανεπιστήμιο Rutgers κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

Το VM παρέχει διαγνωστικές και θεραπευτικές προτάσεις για μετά από εγχείρηση ασθενείς σε τμήμα εντατικής παρακολούθησης (ICU)\*. Το σύστημα αναγνωρίζει πιθανές καταστάσεις προειδοποίησης, αναγνωρίζει πλαστά δεδομένα, χαρακτηρίζει την κατάσταση του ασθενούς και προτείνει χρήσιμες θεραπείες. Το σύστημα ερμηνεύει ποσοτικές μετρήσεις από ένα ICU monitoring system : Τέτοιες μετρήσεις αναφέρονται σε: heart rate, πίεση αίματος και σε δεδομένα που αφορούν έναν μηχανικό ανεμιστήρα που παρέχει στον ασθενή βοήθεια αναπνευστική. Το σύστημα κάνει αυτή την ερμηνεία εφαρμόζοντας γνώσεις για το ιστορικό του ασθενούς και προσδοκίες για τη σειρά των monitored μετρήσεων. Το VM είναι ένα rule-based σύστημα που εκτελείται στην INTERLISP. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο ενός field πρωτοτύπου.

Το WHEEZE κάνει διάγνωση της παρουσίας και σοβαρότητας ασθενείας πνευμόνων ερμηνεύοντας μετρήσεις των τεστ πνευμονικής λειτουργίας. Το σύστημα βασίζει τη διάγνωσή του σε αποτελέσματα κλινικών εργαστηριακών τεστ (π.χ. ολική χωρητικότητα του πνεύμονα) και σε ιστορικά ασθενούς (π.χ. ηλικία, ιστορικό καπνίσματος κ.λ.π.). Του WHEEZE η πραγματογνωμοσύνη αποτελείται από μετάφραση κανόνων που χρησιμοποιούνται από το PUFF σε μια αντιπροσώπευση frame-based. Οι σχηματισμοί περιέχουν δύο τύπους παραγόντων βεβαιότητας. Ο ένας δείχνει την πιθανότητα να είναι σωστός ένας ισχυρισμός και ο άλλος δίνει τον βαθμό με τον οποίο πιστεύεται ότι ο ισχυρισμός και ο άλλος δίνει τον βαθμό με τον οποίο πιστεύεται ότι ο ισχυρισμός αυτός είναι true. Ο μηχανισμός ελέγχου του WHEEZE παρέχει ένα είδος backward και forward chaining που εκτελείται χρησιμοποιώντας μία ατζέντα με κάθε προτεινόμενη βεβαίωση που τοποθετείται στην ημερήσια διάταξη (agenda) σύμφωνα με ειδικευμένη προτεραιότητα. Το WHEEZE εκτελείται στην RLL. Αναπτύχθηκε στο Πανεπιστήμιο Stanford κι έφθασε το στάδιο πρωτότυπης έρευνας.

## 12. METEOROLOGY (Μετεωρολογία).

Το WILLARD βοηθά τους μετεωρολόγους να προβλέπουν την πιθανότητα αυστηρών καταιγίδων που συμβαίνουν στην Κεντρική Αμερική. Το σύστημα ζητά να μάθει από το μετεωρολόγο σχετικά με τις συνθήκες καιρού για την προβλέπυσα περιοχή και μετά παράγει ένα πλήρες δελτίο πρόβλεψης υποστηρίζοντας δικαιολογίες (justifications). Ο χρήστης μπορεί να ειδικεύσει μιά ιδιαίτερη γεωγραφική περιοχή για να μελετήσει το WILLARD. Το σύστημα χαρακτηρίζει τη βεβαιότητα συμβάντος αυστηρής καταιγίδας σαν: "τίποτα", "προσέγγιση", "ελαφρός", "μέτριος" ή "υψηλός" και κάθε ένας από αυτούς τους χαρακτηρισμούς δίνεται με σειρά αριθμητικής πιθανότητας. Η πραγματογνωμοσύνη του WILLARD αντιπροσωπεύεται σαν κανόνες που παράγονται αυτόματα από παραδείγματα προβλεπόντων εμπειρογνωμόνων. Το WILLARD εκτελέσθηκε στην Radia Corporation κι έφθασε το στάδιο επίδειξης πρωτοτύπου.

## 13. Military Science.

Εδώ θα αναφέρουμε απλώς ονομαστικά τα Expert Systems που χρησιμοποιήθηκαν ή χρησιμοποιούνται για πολεμικούς σκοπούς, την μορφή του Knowledge representation, την γλώσσα με την οποία υλοποιήθηκαν και το όνομα του ιδρύματος - γεννήτορά τους.

**ACES:** Χρησιμοποιεί ένα object-oriented Knowledge representation scheme και υλοποιήθηκε σε LOOPS for the Xeros Dolphin workstation. Δημιουργήθηκε στο ESL.

**ADEPT:** Είναι rule-based system και η expert reasoning συνιστώσα του ADEPT υλοποιήθηκε σε ROSIE και χρησιμοποιήθηκε ένα Chromatics CGC 7900 color graphics system για έκθεση χαρτών και military symbology. Δημιουργήθηκε στο TRW.

**AIRID:** Χρησιμοποιεί έναν συνδυασμό rule-based και semantic network τεχνικών αναπαράστασης. Υλοποιήθηκε σε KAS και δημιουργήθηκε at Los Alamos National Laboratory.

**AIRPLAN:** Είναι ένα rule-based system, υλοποιήθηκε σε OPS7 και δημιουργήθηκε στο Carnegie-Mellon University.

**AMUID:** Το expertise του Amuid κωδικοποιείται με την μορφή rules οι οποίοι λειτουργούν on domain knowledge που αναπαρίσταται με την βοήθεια semantic net. Δημιουργήθηκε στο Advanced Information & Decision Systems.

**ANALYST:** Η αναπαράσταση της γνώσης του γίνεται με έναν συνδυασμό rules και frames. Εκτελείται σε FRANZ LISP και δημιουργήθηκε at the Mitre Corporation.

**ASTA:** Η γνώση στο ASTA αναπαρίσταται υπό μορφή κανόνων. Το σύστημα δημιουργήθηκε στο Advanced Information & Decision Systems.

**ART:** Το σύστημα χρησιμοποιεί frames και παράγοντες βεβαιότητας (certainty factors) μαζί με ένα object-oriented control scheme για να αναπαραστήσει την Knowledge. Υλοποιήθηκε σε ZETALISP και αναπτύχθηκε στο Hughes Aircraft Company.

**BATTLE:** Αναπαριστά την γνώση του υπό μορφή rules οι οποίοι συνδέονται με PROSPECTOR-Like certainty values. Δημιουργήθηκε στο Navan-Research Laboratory στην Washington.

**DART:** Το σύστημα υλοποιήθηκε σε PASCAL και C για τον VAX 11/780. Αναπτύχθηκε από την Par Technology Corporation.

**EPES:** Η γνώση στο EPES αναπαρίσταται και με rules-based και semantic net μορφές. Υλοποιήθηκε σε ZETALISP και δημιουργήθηκε στο Texas Instruments.

**EXPERT NAVIGATOR:** Η γνώση στο σύστημα αναπαριστάτε με την βοήθεια “rules” οι οποίοι “λειτουργούν” μέσω μιας blackboard αρχιτεκτονικής. Το EXPERT NAVIGATOR υλοποιήθηκε σε LISP for the symbolics 3600 Workstation. Αναπτύχθηκε στο Analytic Sciences Corporation.

**HANNIBAL:** Η γνώση στο σύστημα αναπαρίσταται μέσω μιας αρχιτεκτονικής blackboard που χρησιμοποιεί πολλαπλές Knowledge-sources. Υλοποιήθηκε σε AGE και αναπτύχθηκε by ESL.

**I & M:** Η γνώση αναπαρίσταται μέσω μιας blackboard αρχιτεκτονικής η οποία χρησιμοποιεί και frames και forward chaining rules για να οργανώσει το expertise. Το σύστημα υλοποιήθηκε σε INTERLISP-D for the Xerox 1100 series workstations. Αναπτύχθηκε από την ESL και το Stanford University.

**KNOBS:** Η γνώση στο KNOBS αναπαρίσταται με frames και backward chaining rules, υλοποιήθηκε σε FRL και ZETALISP. Αναπτύχθηκε στο MITRE Corporation.

**MES:** Το MES είναι ένα forward chaining, rule-based system. Υλοποιήθηκε σε LISP σε έναν Apple II + microcomputer. Αναπτύχθηκε από το Air-Force Institute of Technology.

**OCEAN SURVEILLANCE:** Το σύστημα χρησιμοποιεί ένα rule-based αναπαράστασης γνώσης σχήμα το οποίο “απασχολεί” forward chaining και certainty factors. Υλοποιήθηκε σε FRANZ LISP. Αναπτύχθηκε at Science Applications Inc.

**RTC:** Είναι ένα rule-based system. Υλοποιήθηκε στην FRANZ LISP και αναπτύχθηκε by Advanced Information & Decision Systems.

**RUBRIC:** Η γνώση του αναπαρίσταται με rules. Υλοποιήθηκε σε FRANZ LISP. Αναπτύχθηκε by Advanced Information & Decision Systems.

**SCENARIO-AGENT:** Η γνώση του αναπαρίσταται με rules. Το σύστημα χρησιμοποιεί ένα forward-chaining, rule-based και procedure-oriented inference mechanism. Υλοποιήθηκε in ROSIE και αναπτύχθηκε στο Rand Corporation.

**HASP & SLAP:** Η γνώση του αναπαρίσταται με rules εντός μιας blackboard αρχιτεκτονικής που χρησιμοποιεί ένα hierarchically organized control σχήμα. Το HASP ήταν το αρχικό στάδιο έρευνας πάνω στο οποίο θεμελιώθηκε το SIAP. Το σύστημα υλοποιήθηκε σε INTERLISP από το Stanford University και από το Systems Control Technology.

**SPAM:** Το SPAM υλοποιήθηκε σε ORS5 στο Carnegie-Mellon University.



**SWORL:** Το σύστημα χρησιμοποιεί ένα object-oriented knowledge representation σχήμα. Το σύστημα υλοποιήθηκε σε ROSS και αναπτύχθηκε at the Rand Corporation.

**TART:** Το TART είναι ένα forward-chaining, rule-based system υλοποιημένο σε ROSIE. Αναπτύχθηκε στο Rand Corporation.

**TWIRL:** Το σύστημα χρησιμοποιεί ένα object-oriented knowledge representation σχήμα. Το σύστημα περιλαμβάνει a color graphics facility η οποία παράγει an animated display of the simulation. Το TWIRL υλοποιήθηκε σε ROSS και αναπτύχθηκε από το Rand Corporation.

## 14. ΦΥΣΙΚΗ.

Στον τομέα της Φυσικής (Physics) τα πιο γνωστά EXPERT είναι το GAMMA και το MECHO

## 14. PROCESS CONTROL.

Σ' αυτόν τον τομέα της επιστήμης το πιο γνωστά expert είναι το FALCON και το PDS.

## 16. SPACE TECHNOLOGY

Στον τομέα της τεχνολογίας διαστήματος το πιο γνωστά expert systems κατά αλφαβητική σειρά είναι:

ECESIS, FAITH, KNEECAP, LES, NAVEX, RBMS, RPMS.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- Από την λογική στο λογικό προγραμματισμό και την Prolog.  
Γιώργος Μητακίδης - Εκδόσεις Καρδαμήτσα Αθήνα 1992.
- Έμπειρα συστήματα - σημειώσεις Έρα Αντωνοπούλου 1991
- Buchanam Shirtcliffe rule - based extert system Addision W.P.C 1984.
- Tumer R. lodics for A.I. Ellishorwood series in A.I. Ellishorwood limited 1985.
- Ξανθάκης Σ. prolog, Τεχνικές προγραμματισμού, εκδόσεις νέων τεχνολογιών Αθήνα 1990
- Μητακίδης Γ. Μαθηματική Λογική Πανεπιστήμια Πατρών 1985