

## ΑΣΚΗΣΗ 4

Πρωταρχική μέθοδος αναπαράστασης γνώσης στο CLIPS είναι οι κανόνες. Κάθε κανόνας καθορίζει ένα σύνολο ενεργειών που πρέπει να γίνουν δεδομένης μιας κατάστασης. Για την επίλυση ενός προβλήματος χρειάζεται να οριστεί ένα σύνολο κανόνων που λειτουργούν συλλογικά για το σκοπό αυτό. Στόχος της τέταρτης εργαστηριακής άσκησης είναι η **Δημιουργία Κανόνων**.

### Κανόνες στο CLIPS

- Ένας κανόνας, όπως ήδη ξέρετε, αποτελείται από δύο μέρη: το προηγούμενο-προϋπόθεση (**antecedent**) και το επακόλουθο (**consequent**) που ονομάζονται και Left-Hand Side (**LHS**), Right-Hand Side (**RHS**), αντίστοιχα. Πρόκειται ουσιαστικά για τα μέρη **if** και **then** ενός κανόνα. Ανοίξτε το CLIPS και τα παράθυρα Facts και Agenda. Ορίστε τους παρακάτω κανόνες:

1. (defrule rule1 "find fruit from colour"  
     (colour red)  
     =>  
     (assert (fruit strawberry)))
2. (defrule rule2 "find fruit from colour"  
     (colour yellow)  
     =>  
     (assert (fruit banana)))
3. (defrule rule3 "find fruit from colour"  
     (colour orange)  
     =>  
     (assert (fruit orange)))
4. (defrule rule4 "who plays the instrument"  
     (instrument classical-guitar)  
     =>  
     (assert (name Xrhstos)))
5. (defrule rule5 "who plays the instrument"  
     (instrument piano)  
     =>  
     (assert (name Panagiotis)))
6. (defrule rule6 "who plays the instrument"  
     (instrument electric-guitar)  
     =>  
     (assert (name Xrhstos)))

Εξηγήστε τι κάνει κάθε κανόνας. Τι αποτελεί το LHS και το RHS καθενός από τους παραπάνω κανόνες; Παρατηρήστε πως διαχωρίζονται τα δύο μέρη ενός κανόνα μεταξύ τους. Ο ορισμός ενός κανόνα, όπως βλέπετε, γίνεται μέσω της συνάρτησης **defrule**. Με την εντολή (**rules**) εμφανίστε τους κανόνες που

δημιουργήσατε. Έστω πως θέλουμε να σώσουμε τους κανόνες που ορίσαμε στο CLIPS σε ένα αρχείο για να μπορούμε να τους ξαναφορτώσουμε όποτε χρειαστεί χωρίς να τους ξαναγράψουμε. Χρησιμοποιήστε την εντολή **save** για να σώσετε τους ορισμούς σας στο αρχείο `rules.clp`:

(save "rules.clp")

Ανοίξτε το αρχείο `rule.clp` και παρατηρήστε την μορφή στην οποία αποθηκεύονται οι κανόνες σε αυτό.

- Από το menu **Browse** επιλέξτε **Defrule Manager...** για να εξετάσετε τους κανόνες που μόλις δημιουργήσατε. Επιλέξτε τον κανόνα `rule6` και διαγράψτε τον. Δείτε στο **Dialog Window** με ποια εντολή ισοδυναμεί το **Remove** του **Defrule Manager**. Χρησιμοποιήστε αυτήν την εντολή για να διαγράψετε και τον κανόνα `rule1` και επιβεβαιώστε πως ο κανόνας διαγράφηκε από την λίστα του **Defrule Manager**. Επιλέξτε τον κανόνα `rule4` και πατήστε **Pprint**. Με ποια εντολή ισοδυναμεί αυτή η ενέργεια; Χρησιμοποιήστε αυτήν την εντολή για να εμφανίσετε το κείμενο όλων των υπολοίπων κανόνων. Όταν εκτυπώνονται οι κανόνες, τι εμφανίζεται μπροστά από το όνομά τους και τι σημαίνει αυτό;
- Δοκιμάστε τώρα να ορίσετε τους παρακάτω κανόνες:

```
1. (defrule rule2
    "find fruit from colour"
    (colour red)
    =>
    (assert (fruit cherry)))
```

Επιλέξτε **Browse-> Defrule Manager...** και δείτε πόσοι κανόνες με το όνομα `rule2` υπάρχουν στην λίστα. Στη συνέχεια εκτυπώστε τον κανόνα `rule2` με οποιαδήποτε από τις δύο δυνατές επιλογές που είδατε πιο πάνω. Επαναλάβετε την ίδια διαδικασία για τον επόμενο κανόνα:

```
2. (defrule rule2
    "find fruit from colour"
    (colour red)
    =>
    (assert (fruit apple)))
```

Τι παρατηρείτε; Τι συμβαίνει όταν ορίζουμε έναν νέο κανόνα χρησιμοποιώντας το όνομα ενός ήδη υπάρχοντος κανόνα; Αυτή τη στιγμή ποιος κανόνας `rule2` υπάρχει στους κανόνες μας. Κάντε ό, τι χρειάζεται ώστε να εισάγετε και τους άλλους δύο κανόνες του ίδιου ονόματος που χάθηκαν. Επιβεβαιώστε πως μετά από αυτήν την διαδικασία υπάρχουν συνολικά 6 κανόνες ξανά στην λίστα κανόνων.

- Δοκιμάστε να δώσετε την εντολή (`reset`) στο **Dialog Window**. Μετά δώστε την εντολή (`run`). Τι συμβαίνει; Στην συνέχεια, ορίστε τα παρακάτω διατεταγμένα γεγονότα στο CLIPS:

```
1. (deffacts some-facts
```

(colour orange)  
(instrument piano))

Δώστε την εντολή (reset) μετά το prompt. Παρατηρήστε τα περιεχόμενα των παραθύρων Facts και Agenda. Τι δηλώνει κάθε γραμμή στο παράθυρο Agenda; Ποιοι είναι οι ενεργοποιημένοι κανόνες; Πού οφείλεται η ενεργοποίησή τους και η μη ενεργοποίηση των υπόλοιπων κανόνων; Τι σημαίνει όταν λέμε πως ένας κανόνας έχει ενεργοποιηθεί; Εκτυπώστε με την εντολή **ppdefrule** τους ενεργοποιημένους κανόνες. Ο αριθμός μπροστά από το όνομα κάθε ενεργοποιημένου κανόνα στο παράθυρο Agenda δηλώνει την προτεραιότητά του σε σχέση με τους υπόλοιπους κατά την εκτέλεση. Συνεπώς, οι αριθμοί μπροστά από τα ονόματα των ενεργοποιημένων κανόνων δηλώνουν την σειρά με την οποία αυτοί θα εκτελεστούν. Ποιο περιμένετε να είναι το αποτέλεσμα μετά την εκτέλεση των ενεργοποιημένων κανόνων; Γιατί δεν έχει επηρεαστεί ακόμα η λίστα γεγονότων από τους ενεργοποιημένους κανόνες, αφού οι συνθήκες τους ικανοποιούνται; Δώστε την εντολή (run) και παρατηρήστε τις αλλαγές στα παράθυρα Facts και Agenda. Τα αποτελέσματα συμφωνούν με αυτά που περιμένατε;

Να θυμάστε πως η εντολή (**run**) είναι απαραίτητη για την εκτέλεση των ενεργοποιημένων κανόνων.

Το CLIPS εκτελεί πάντα τον κανόνα με την μεγαλύτερη προτεραιότητα ενεργοποίησης. Αφού εκτελεστεί ο κανόνας αυτός, διαγράφεται από την στοίβα ενεργοποιημένων κανόνων (agenda) κι εκτελείται ο επόμενος, αμέσως μικρότερης (ή ίσης) προτεραιότητας ενεργοποιημένος κανόνας.

Η διαδικασία αυτή συνεχίζεται είτε μέχρι την στιγμή όπου δεν υπάρχουν πλέον άλλοι ενεργοποιημένοι κανόνες προς εκτέλεση στην agenda, είτε μέχρι να βρεθεί κάποια εντολή για λήξη της εκτέλεσης.

Τώρα ορίστε τα παρακάτω γεγονότα. Χρησιμοποιείται το ίδιο όνομα με την προηγούμενη δομή deffacts, έτσι ώστε μετά την νέα εντολή (reset) να αντικατασταθούν τα προηγούμενα γεγονότα από τα καινούργια.

2. (defacts some-facts  
(color red)  
(instrument classical-guitar))

Δώστε ξανά την εντολή (reset) και αφού φορτωθούν οι ορισμοί στο CLIPS παρατηρήστε το παράθυρο Agenda. Ποιο το περιεχόμενό του; Γιατί δεν υπάρχει κάποιος ενεργοποιημένος κανόνας; Βρείτε τι φταίει και διορθώστε το. Στην συνέχεια δείτε στην Agenda τους ενεργοποιημένους κανόνες και εκτελέστε τους μέσω της εντολής (run). Με την εντολή retract διαγράψτε τα γεγονότα που προστέθηκαν στην λίστα γεγονότων από την εκτέλεση των κανόνων. Αφήστε μόνο τα τρία πρώτα γεγονότα. Δοκιμάστε τώρα να δώσετε ξανά την εντολή (run). Τι συμβαίνει; Γιατί δεν εκτελούνται ξανά οι κανόνες; Το φαινόμενο αυτό ονομάζεται **refraction** και βοηθάει στην αποφυγή ατέρμονων βρόχων στις περιπτώσεις όπου ένας κανόνας εκτελείται και εξακολουθούν να υπάρχουν στην λίστα γεγονότων τα γεγονότα εκείνα που τον ικανοποιούν.

➤ Δοκιμάστε να κάνετε τα εξής:

1. (watch activations)
2. (retract \*)
3. (agenda)
  
4. (assert (colour red))
5. (agenda)
  
6. (assert (instrument classical-guitar))
7. (agenda)
  
8. (run 1)
9. (agenda)
  
10. (run 2)
11. (agenda)
  
12. (run 2)
13. (agenda)

Τι κάνει η εντολή (**watch activations**); Τι κάνει η εντολή (**agenda**); Τι δηλώνει το όρισμα που δίνουμε στην εντολή run; Τι είναι αυτό που κάναμε εδώ και μπόρεσαν να ενεργοποιηθούν και να εκτελεστούν ξανά οι προηγούμενοι κανόνες;

Ένας κανόνας ενεργοποιείται όταν οι συνθήκες του ικανοποιούνται:

- από ένα νέο γεγονός που δεν υπήρχε πριν, ή
- από ένα γεγονός που υπήρχε πριν αλλά διαγράφηκε και εισήχθη ξανά στην λίστα γεγονότων, οπότε θεωρείται αντίγραφο του προηγούμενου γεγονότος και συνεπώς, νέο γεγονός.

- Θα δούμε έναν εναλλακτικό τρόπο να αναγκάσουμε κάποιον κανόνα να ενεργοποιηθεί εκ νέου από τα ίδια τα γεγονότα που τον έχουν ενεργοποιήσει ήδη μια φορά. Διαγράψτε από την λίστα γεγονότων όλα τα γεγονότα εκτός από τα: (colour red) και (instrument classical-guitar). Αν τώρα δώσετε την εντολή (run) δεν θα συμβεί τίποτα. Δοκιμάστε να κάνετε το εξής: Επιλέξτε **Browse->Defrule Manager...** και πατήστε το κουμπί **Refresh** για τον κανόνα rule1. Κλείστε το παράθυρο διαλόγου του Defrule Manager και παρατηρήστε το παράθυρο Agenda. Παρατηρήστε σε ποια εντολή αντιστοιχεί η ενέργεια Refresh στο Dialog Window και χρησιμοποιήστε την για να ανανεώσετε και τους υπόλοιπους κανόνες. Μετά την ανανέωση κάθε κανόνα παρατηρήστε τι συμβαίνει στο παράθυρο Agenda. Δώστε ξανά την εντολή (run). Τι κάνει η εντολή refresh; Απενεργοποιήστε την παρακολούθηση των ενεργοποιήσεων με την εντολή (**unwatch activations**).
  
- Το μέρος LHS ενός κανόνα αποτελείται από τις συνθήκες του οι οποίες πρέπει να ικανοποιηθούν προκειμένου να ενεργοποιηθεί ο κανόνας. Μια συνθήκη ικανοποιείται όταν μπορεί να ενοποιηθεί/ταιριάξει με κάποιο γεγονός της λίστας γεγονότων. Όπως αναφέρεται και στις σημειώσεις του μαθήματος, η

διαδικασία της ενοποίησης/ταιριάσματος (**matching**) έχει σαν στόχο να βρει αν υπάρχει κάποιο γεγονός με το οποίο μπορεί να ταυτοποιηθεί μια συνθήκη ενός κανόνα. Υπάρχουν οι εξής δύο περιπτώσεις:

- Η συνθήκη του κανόνα δεν περιέχει μεταβλητές αλλά κυριολεκτικά (**literals**).
- Η συνθήκη του κανόνα περιέχει μεταβλητές ενός ή πολλαπλών πεδίων.

Στην πρώτη περίπτωση η συνθήκη ικανοποιείται μόνο από κάποιο γεγονός πανομοιότυπο με αυτήν. Στην δεύτερη περίπτωση η συνθήκη ικανοποιείται από όλα τα γεγονότα που μπορούν να γίνουν πανομοιότυπα με αυτήν αν δοθούν συγκεκριμένες τιμές στις μεταβλητές της. Επιπλέον, διακρίνουμε τις συνθήκες σε: **Συνθήκες Προτύπου**, **Συνθήκες Διεύθυνσης Προτύπου**, **Εκτιμήσιμες Συνθήκες** και **Λογικά Συνδεόμενες Συνθήκες**. Εδώ μας ενδιαφέρουν μόνο οι συνθήκες προτύπου και οι συνθήκες διεύθυνσης προτύπου. Με τις υπόλοιπες θα ασχοληθούμε στις επόμενες ασκήσεις. Δώστε την εντολή (clear), ορίστε τους παρακάτω κανόνες σε ένα αρχείο και φορτώστε το αρχείο αυτό στο CLIPS:

1. (defrule match1  
    (colours red black white orange)  
    =>)
2. (defrule match2  
    (colours red ?x white ?y)  
    =>)
3. (defrule match3  
    (colours red ?x ?y ?z)  
    =>)
4. (defrule match4  
    (colours red ? ? ?)  
    =>)
5. (defrule match5  
    (colours \$? blue)  
    =>)
6. (defrule match6  
    (colours red ?x white ?x)  
    =>)
7. (defrule match7  
    (colours \$?rest blue)  
    =>)

Για κάθε γεγονός από τα παρακάτω σημειώστε δίπλα το όνομα (ή τα ονόματα) του κανόνα (ή των κανόνων) που περιμένετε να ενεργοποιηθεί (ενεργοποιηθούν) από αυτό. Με άλλα λόγια, βρείτε τους κανόνες των οποίων οι συνθήκες ενοποιούνται με κάθε γεγονός:

- (colour red black white orange)
- (colors red blue white white)
- (colours red black white orange)
- (colours red)
- (colours purple black white orange)
- (colours purple black red blue)
- (colours red black white black)
- (colours red black white blue)
- (colours red blue)
- (colours white black purple brown green orange yellow blue)

Εισάγετε τώρα ένα-ένα τα παραπάνω γεγονότα και επιβεβαιώστε ή διαψεύστε τις απαντήσεις σας. Για να διευκολυνθείτε δίνετε την εντολή (retract \*) μεταξύ της προηγούμενης και της επόμενης εισαγωγής. Ποια γεγονότα δεν ενοποιούνται με καμία συνθήκη και γιατί; Ποιος ο ρόλος των wildcards ? και \$? στις συνθήκες των κανόνων 4 και 5; Ποια η διαφορά των κανόνων αυτών με τους 3 και 7 όπου έχουμε μεταβλητές μίας και πολλαπλών τιμών αντίστοιχα; Πόσα είναι τα πεδία που πρέπει να έχει απαραίτητα ένα γεγονός για να μπορέσει να ενεργοποιήσει τον κανόνα 4; Πόσα είναι τα πεδία που πρέπει να έχει ένα γεγονός για να μπορέσει να ενεργοποιήσει τον κανόνα 5; Ποια η διαφορά των κανόνων 2 και 6; Ποια είναι η διαφορά που παρατηρείτε όσον αφορά την ενεργοποίηση κανόνων με σταθερά πεδία στις συνθήκες τους και την ενεργοποίηση κανόνων με μεταβλητές στις συνθήκες τους;

- Οι συνθήκες των κανόνων που έχουμε δει ως τώρα είναι συνθήκες προτύπου (**pattern**). Όπως αναφέρεται και στις σημειώσεις του μαθήματος, το πρώτο πεδίο κάθε τέτοιας συνθήκης είναι πάντα ένα σύμβολο. Τι δηλώνει το σύμβολο αυτό; Τα υπόλοιπα πεδία είτε έχουν σταθερές τιμές, είτε είναι μεταβλητές, είτε wildcards, είτε κάποιος περιορισμοί (λογικής σύνδεσης, κατηγορήματος, επιστροφόμενης τιμής). Τα πεδία μιας συνθήκης καθορίζουν τις τιμές που πρέπει να έχει στα αντίστοιχα πεδία του ένα γεγονός, προκειμένου να μπορεί να ενοποιηθεί με την συγκεκριμένη συνθήκη. Προσθέστε τώρα τους παρακάτω ορισμούς στο τέλος του αρχείου που έχετε ήδη δημιουργήσει. Σώστε το αρχείο σας ξανά, δώστε την εντολή (clear) και φορτώστε το αλλαγμένο αρχείο στο CLIPS.

- ```
8. (defrule match8
  ?p1 <- (person ?name ?surname $?kids)
  ?p2 <- (person ?name2 ?surname $?kids)
  =>
  (assert (kids of ?name ?name2 ?kids))
  (retract ?p1 ?p2))

9. (defrule match9
  (colours red ~red ~red ~red)
  =>)

10. (defrule match10
  (colours red ?x&~pink&~black white orange)
  =>)
```

11. (defrule match11
 

```
?p <- (person $?name male | female)
=>
(assert (human ?name))
(retract ?p))
```
12. (defrule match12
 

```
?p <- (person $?name female)
=>
(assert (woman ?name))
(retract ?p))
```
13. (defrule match13
 

```
(data ?x&:(numberp ?x))
=>
(printout t "executing match13..." crlf))
```
14. (defrule match14
 

```
(data ?x&:(numberp ?x)&:(oddp ?x))
=>
(printout t "executing match14..." crlf))
```
15. (defrule match15
 

```
(data ?y)
(data ?x&:(> ?x ?y))
=>
(printout t "executing match15..." crlf))
```
16. (defrule match16
 

```
(data ?x ?y&=(2 ?x))
=>
(printout t "executing match16..." crlf)
(assert (result ?y)))
```
17. (defrule match17
 

```
(numbers ?x ?y)
(hypotenuse ?h&=(sqrt (+ (** ?x 2) (** ?y 2))))
=>
(printout t "executing match17..." crlf)
(assert (hypotenuse ?h)))
```

Πως ονομάζεται ο τύπος των συνθηκών του κανόνα 8; Ποιος ο ρόλος αυτού του τύπου συνθηκών; Σε ποιους άλλους κανόνες εμφανίζονται τέτοιες συνθήκες και σε τι χρησιμεύουν; Δοκιμάστε να κάνετε τα παρακάτω:

- (assert (person Helen Johnson Peter Mary Chris))
- (assert (person James Johnson Peter Mary Chris))
- (run 1)
- (run 1)

Τι δεν πάει καλά; Επαναλάβετε τις παραπάνω ενέργειες αφού πρώτα αλλάξετε την συνθήκη: ?p2 <- (person ?name2 ?surname \$?kids) με την συνθήκη:

?p2 <- (person ?name2&~?name ?surname \$?kids). Διορθώθηκε το πρόβλημα; Τι χρησιμοποιήσαμε εδώ για να διορθώσουμε το πρόβλημα; Κάνετε τις παρακάτω ενέργειες παρατηρώντας τις ενεργοποιήσεις που συμβαίνουν:

- (assert (person George Jackson Peter Mary Chris))
- (assert (person James Johnson Peter Mary))
- (assert (person James Johnson Peter Mary Chris))
- (assert (person Jim Robinson Alex))
- (assert (person Olina Robinson Alex))
- (assert (person Olina Robinson Bill))

Παρατηρήστε τα διαφορετικά στιγμιότυπα του ίδιου κανόνα που ενεργοποιούνται για κάθε διαφορετικό συνδυασμό γεγονότων που ικανοποιεί τις συνθήκες του. Τώρα δώστε την εντολή (run) για να εκτελεστούν τα ενεργοποιημένα στιγμιότυπα του κανόνα.

Ποιους κανόνες περιμένετε να ενεργοποιήσουν τα παρακάτω γεγονότα; Εισάγετε τα στην λίστα γεγονότων και παρατηρήστε ποιοι κανόνες ενεργοποιούνται κάθε φορά. Μετά από καθεμία από τις πρώτες πέντε εισαγωγές δίνετε την εντολή (retract \*) προκειμένου να βλέπετε στην agenda μόνο τους κανόνες που ενεργοποιούνται από την τελευταία εισαγωγή.

- (colours red red green blue)
- (colours red orange yellow black)
- (colours red pink white orange)
- (colours red blue white orange)
- (colours black blue white orange)
- (person Anna Polith female)
- (person Alex Johnson male)

Μετά τις δύο τελευταίες εισαγωγές δώστε την εντολή (run) και παρατηρήστε τα αποτελέσματα. Γιατί ενώ ενεργοποιούνται δύο κανόνες από το γεγονός (person Anna Polith female), εκτελείται μονάχα ο ένας; Αλλάξτε τους κανόνες match11 και match12 έτσι ώστε κατά την εκτέλεσή τους να μην διαγράφεται το γεγονός person που τους ενεργοποιεί. Δώστε την εντολή (clear) και φορτώστε ξανά το αρχείο στο CLIPS. Εισάγετε τα τελευταία δύο γεγονότα ξανά και δώστε την εντολή (run). Τι γίνεται τώρα;

- Ορίστε έναν νέο κανόνα στο CLIPS που να κάνει ότι και ο match12 αλλά για άτομα αρσενικού φύλου. Δώστε την εντολή (clear) και φορτώστε ξανά το αρχείο στο CLIPS. Εισάγετε ξανά τα δύο τελευταία γεγονότα και δώστε την εντολή (run). Αφού δείτε τα αποτελέσματα δώστε την εντολή (retract \*) και συνεχίστε με τις παρακάτω εισαγωγές γεγονότων. Ποιοι κανόνες περιμένετε να εκτελεστούν μετά την εντολή (run) και πόσες φορές ο καθένας;

- (assert (data 1))
- (assert (data red blue 3))
- (assert (data 2 3 4))
- (assert (data 2))
- (assert (data 3))



- (assert (data 0))
- (assert (data 1 2))
- (assert (data 2 4))
- (assert (data 0 1))
- (assert (numbers 3 4))
- (assert (hypotenuse 5.0))
- (assert (numbers 3))
- (run)

Ποιοι από τους κανόνες 1-17 έχουν στις συνθήκες τους **περιορισμούς λογικής σύνδεσης**, ποιοι έχουν **περιορισμούς κατηγορήματος** και ποιοι **περιορισμούς επιστρεφόμενης τιμής**; Ποιος ο ρόλος καθενός από αυτούς τους περιορισμούς;

➤ Δώστε τις παρακάτω εντολές στο CLIPS:

1. (matches match17)
2. (matches match16)
3. (matches match14)
4. (matches match13)
5. (matches match1)
6. (undefrule \*)
7. (rules)

Τι κάνει η εντολή **matches** και τι δέχεται σαν όρισμα; Εναλλακτικά, πως μπορούμε να κάνουμε το ίδιο πράγμα χωρίς να δώσουμε την εντολή matches οι ίδιοι; (Θυμηθείτε τις εντολές undefrule, refresh και rredefrule). Τι κάνει η εντολή 6;

➤ Κλείστε το CLIPS είτε με την εντολή (exit) στο Dialog Window είτε με File->Exit.