

ΑΣΚΗΣΗ 3

Ένα γεγονός είναι μια λίστα από σύμβολα τα οποία περικλείονται σε παρενθέσεις και εκφράζει κάποια πληροφορία σχετική με την κατάσταση του προβλήματος που μας απασχολεί. Τα γεγονότα διακρίνονται σε δύο κατηγορίες: τα Διατεταγμένα Γεγονότα και τα Μη Διατεταγμένα Γεγονότα. Στόχος της τρίτης εργαστηριακής άσκησης είναι η χρήση **Μη Διατεταγμένων Γεγονότων** ή αλλιώς η χρήση **Προτύπων Γεγονότων**.

Μη Διατεταγμένα Γεγονότα στο CLIPS

➤ Παρατηρήστε τα παρακάτω γεγονότα:

1. (person Maria Papadopoulou 42 "T234657" 1.65 teacher Athens)
2. (person "T234657" Papadopoulou Maria 1.65 Athens teacher)
3. (cat Tom black)
4. (family (father John) (mother Lily) (kids Michael George Bill))
5. (hotel (location Patras) (name "Astir") (address "Ag.Andreou 16") (facilities pool sauna parking internet-corner roof-garden))
6. (family John Lily Michael George Bill)
7. (person (name Maria) (surname Papadopoulou) (id "T234657") (height 1.65) (profession teacher) (home Athens))
8. (cat (name Tom) (colour black))
9. (family (kids Michael George Bill) (mother Lily) (father John))
10. (family (parents John Lily) (kids Michael George Bill))
11. (person (id "T234657") (surname Papadopoulou) (name Maria) (height 1.65) (home Athens) (profession teacher))
12. (hotel Patras "Astir" "Ag.Andreou 16" pool sauna parking internet-corner roof-garden)

Σημειώστε ποια από τα παραπάνω γεγονότα είναι διατεταγμένα γεγονότα και ποια προέρχονται από κάποιο πρότυπο και είναι μη διατεταγμένα γεγονότα. Ομαδοποιήστε διατεταγμένα και μη διατεταγμένα γεγονότα που εκφράζουν ακριβώς την ίδια κατάσταση-πληροφορία. Ποια είναι η διαφορά που παρατηρείτε στα πεδία ενός διατεταγμένου γεγονότος και στα πεδία ενός μη διατεταγμένου γεγονότος; Από τι διακρίνουμε ένα πεδίο σε ένα διατεταγμένο γεγονός και από τι σε ένα μη διατεταγμένο γεγονός; Σημειώστε δίπλα στις παρακάτω προτάσεις αν είναι Σωστές ή Λάθος:

1. Αν κάνουμε assert το γεγονός 1 και το γεγονός 2 εισάγεται το ίδιο γεγονός στο CLIPS.
2. Αν κάνουμε assert το γεγονός 7 και το γεγονός 11 εισάγεται το ίδιο γεγονός στο CLIPS.
3. Αν κάνουμε assert το γεγονός 4 και το γεγονός 9 εισάγεται το ίδιο γεγονός στο CLIPS.
4. Αν κάνουμε assert το γεγονός 9 και το γεγονός 10 εισάγεται το ίδιο γεγονός στο CLIPS.
5. Αν κάνουμε assert το γεγονός 5 και το γεγονός 12 εισάγεται το ίδιο γεγονός στο CLIPS.

Για κάθε πρόταση που θεωρήσατε Λάθος δικαιολογήστε την απάντησή σας. Προσπαθήστε τώρα να εισάγετε στην λίστα γεγονότων διαδοχικά τα γεγονότα 3 και 8. Τι παρατηρείτε; Γιατί το CLIPS εμφανίζει αυτό το μήνυμα; Τι είναι απαραίτητο να γίνει ώστε να μπορεί να ολοκληρωθεί με επιτυχία και η εντολή assert που αφορά το γεγονός 8;

- Ένα Μη Διατεταγμένο Γεγονός, όπως θα παρατηρήσατε, είναι μια λίστα ονοματισμένων πεδίων (named fields), καθένα από τα οποία ονομάζεται **slot**. Ένα slot μπορεί να είναι είτε ονοματισμένο single-slot (ή απλά **slot**), το οποίο περιέχει ακριβώς ένα πεδίο, είτε ονοματισμένο **multislot**, το οποίο περιέχει μηδέν ή περισσότερα πεδία. Για κάθε μη διατεταγμένο γεγονός που σημειώσατε στην προηγούμενη άσκηση αναφέρατε τα ονόματα από όλα τα slots του και διακρίνετε ποια από αυτά είναι single-slots και ποια multislots.
- Όλα τα μη διατεταγμένα γεγονότα που έχουν κοινό το πρώτο τους πεδίο και έχουν ακριβώς τα ίδια slots/multislots προκύπτουν από το ίδιο Πρότυπο Γεγονότος (**Template**), γι' αυτό λέγονται και Γεγονότα Προτύπου. Ένα Πρότυπο Γεγονότων είναι ένας ορισμός που περιγράφει τα πεδία που θα έχουν όλα τα μη διατεταγμένα γεγονότα που θα προκύψουν από αυτό. **Προκειμένου να δημιουργήσουμε, να εισάγουμε στην λίστα γεγονότων και να χειριστούμε μη διατεταγμένα γεγονότα είναι απαραίτητο να ορίσουμε πρώτα κατάλληλα πρότυπα που θα περιγράφουν αυτά τα γεγονότα**, αλλιώς το CLIPS συμπεριφέρεται σαν να εισάγουμε διατεταγμένο γεγονός με λάθος σύνταξη. Ένα πρότυπο για μη διατεταγμένα γεγονότα ορίζεται μέσω της εντολής **deftemplate**. Προσπαθήστε να αντιστοιχίσετε τα παρακάτω πρότυπα (1-4) με τα κατάλληλα μη διατεταγμένα γεγονότα (a-d):

1. (deftemplate hotel
 (slot location)
 (slot name)
 (slot address)
 (multislot facilities))
2. (deftemplate person
 (slot name)
 (slot surname)
 (slot id)
 (slot height)
 (slot profession))

(slot home))

3. (deftemplate family
(multislot parents)
(multislot kids))
 4. (deftemplate cat
(slot name)
(slot colour))
- a. (person (name Maria) (surname Papadopoulou) (id "T234657")
(height 1.65) (profession teacher) (home Athens))
 - b. (cat (name Tom) (colour black))
 - c. (family (parents John Lily) (kids Michael George Bill))
 - d. (hotel (location Patras) (name "Astir") (address "Ag.Andreou
16") (facilities pool sauna parking internet-corner roof-garden))

Ορίστε στο CLIPS τα πρότυπα γεγονότων 1-4 και στην συνέχεια κάνετε assert τα παρακάτω γεγονότα. Μετά από κάθε εισαγωγή ελέγχετε τα γεγονότα στη λίστα γεγονότων με την εντολή (facts):

1. (cat (name Tom) (colour black))
2. (family (parents Tim Helen) (kids Alex))
3. (family (kids Nick Lina Marina) (parents John Mary))
4. (cat (colour white) (name Sissy))
5. (person (name Mary) (surname Papadopoulou))
6. (person (id "T245699"))
7. (hotel (name "Astir") (facilities internet-corner sauna parking roof-garden))

Τι παρατηρείτε σχετικά με το όνομα ενός προτύπου και το πρώτο πεδίο όλων των μη διατεταγμένων γεγονότων που ανταποκρίνονται στο πρότυπο αυτό; Μπορούμε να εισάγουμε μη διατεταγμένα γεγονότα κάποιου προτύπου με οποιαδήποτε σειρά των πεδίων τους; Σε ποιες από τις παραπάνω εισαγωγές φαίνεται αυτό; Τα μη διατεταγμένα γεγονότα που εισάγουμε στη λίστα γεγονότων έχουν την ίδια σειρά πεδίων με αυτήν που χρησιμοποιήσαμε για να τα εισάγουμε; Από τι καθορίζεται η σειρά των πεδίων ενός μη διατεταγμένου γεγονότος όταν εισάγεται στην λίστα γεγονότων; Όταν εισάγουμε ένα μη διατεταγμένο γεγονός είναι ανάγκη να δώσουμε τιμές σε όλα τα πεδία του; Ισχύει το ίδιο για ένα διατεταγμένο γεγονός;

- Για κάθε πεδίο ενός μη διατεταγμένου γεγονότος μπορούμε να θέσουμε ένα σύνολο περιορισμών (constraint forms) μέσα από τον ορισμό deftemplate του αντίστοιχου προτύπου. Αφού δώσετε την εντολή (clear) στο CLIPS, δημιουργήστε ένα αρχείο templates.clp μέσα στο οποίο θα ορίσετε τα παρακάτω πρότυπα μη διατεταγμένων γεγονότων:

```
(deftemplate hotel
  (slot location (type SYMBOL) (default ?NONE))
  (slot name (type STRING) (default ?NONE))
  (slot address (type STRING))
  (multislot facilities (type SYMBOL)
    (allowed-symbols pool sauna parking internet-corner)))

(deftemplate person
  (slot name (type SYMBOL) (default ?NONE))
  (slot surname (type SYMBOL) (default ?NONE))
  (slot id (type STRING)(default ?DERIVE))
  (slot height (type FLOAT) (default 1.65))
  (slot profession (type SYMBOL))
  (slot home(type SYMBOL)))

(deftemplate family
  (multislot parents (type SYMBOL) (cardinality 2 2))
  (multislot kids (type SYMBOL) (cardinality 1 15)))

(deftemplate cat
  (slot name (type SYMBOL) (default ?NONE))
  (slot colour (type SYMBOL) (default unknown)))
```

Με την επιλογή **Deftemplate Manager...** από το menu **Browse** μπορούμε να δούμε και να διαχειριστούμε τα πρότυπα γεγονότων που έχουμε ορίσει με τρόπο ανάλογο με αυτόν που παρουσιάσαμε στην προηγούμενη άσκηση για τους ορισμούς deffacts. Πριν συνεχίσετε με τα επόμενα, κάνετε assert τα διατεταγμένα γεγονότα (dog Jack) και (animal duck). Παρατηρήστε μέσω του Deftemplate Manager ότι για κάθε διατεταγμένο γεγονός που εισάγουμε δημιουργείται αυτόματα από το CLIPS ένα μη διατεταγμένο γεγονός με όνομα ίδιο με το πρώτο πεδίο του διατεταγμένου γεγονότος και ένα μοναδικό slot/multislot το οποίο περιέχει τα υπόλοιπα πεδία του. Δώστε την εντολή (clear) και φορτώστε το αρχείο templates.clp στο CLIPS. Κάνετε τώρα assert τα παρακάτω γεγονότα, έχοντας ανοιχτό το παράθυρο Facts και παρατηρώντας τα μηνύματα που εμφανίζει το CLIPS για κάθε λανθασμένη εισαγωγή:

1. (cat (colour black))
2. (cat (name Kitty))
3. (hotel (address "Ag.Andreou 16")(facilities pool sauna))
4. (hotel (location Patras) (name "Astir") (address "Ag.Andreou 16") (facilities pool sauna))
5. (cat (name "Kitty"))
6. (hotel (location Patras) (name "Astir") (address "Ag.Andreou 16") (facilities breakfast-in-room))
7. (person (name Maria) (surname Papadopoulou) (profession teacher) (home Patras))
8. (family (parents John) (kids Peter))
9. (family (parents John Helen) (kids))
10. (family (parents John Helen) (kids Peter))
11. (family (parents John Helen) (kids Peter))

12. (family (parents Helen John) (kids Peter))

Ποιες εισαγωγές από τις παραπάνω εμφάνισαν μήνυμα λάθους και γιατί; Τι παρατηρήσατε στο παράθυρο Facts μετά την εισαγωγή 11 και μετά την εισαγωγή 12. Σε ποια πεδία ενός μη διατεταγμένου γεγονότος πρέπει να δώσουμε οπωοδήποτε τιμή για να εισαχθεί αυτό στην λίστα γεγονότων επιτυχώς; Τι γίνεται όταν εισάγεται στη λίστα γεγονότων ένα γεγονός χωρίς να έχουμε δώσει τιμές σε όλα τα slots του; Ποιοι είναι οι περιορισμοί που εφαρμόζονται στα πεδία των παραπάνω προτύπων και πώς συντάσσονται; Τι περιορισμούς εισάγουν τα παρακάτω:

- (type SYMBOL)
- (allowed-symbols pool sauna parking)
- (range 2 20)
- (range 0 ?x)
- (range ?x 100)
- (default “unknown”)
- (default ?DERIVE)
- (default ?NONE)
- (cardinality 0 3)
- (cardinality 2 2)

Ποιοι περιορισμοί είναι άμεσα εξαρτώμενοι από τον περιορισμό τύπου type; Ποιοι περιορισμοί αφορούν μόνο multislot πεδία; Μπορούμε να χρησιμοποιήσουμε ταυτόχρονα τον περιορισμό range με οποιονδήποτε από τους περιορισμούς allowed-values, allowed-integers, allowed-floats ή allowed-numbers; Γιατί;

- Έκτος από τις εντολές **assert** και **retract** (για εισαγωγή και διαγραφή γεγονότων αντιστοιχα) μια ακόμα εντολή που μπορούμε να εφαρμόσουμε σε μη διατεταγμένα γεγονότα είναι η εντολή **modify**, η οποία αλλάζει τιμές σε συγκεκριμένα πεδία ενός μη διατεταγμένου γεγονότος. Όπως είπαμε και πιο πάνω, τα πρότυπα γεγονότων απλοποιούν πολύ την πρόσβαση σε κάποιο συγκεκριμένο πεδίο γεγονότος, αφού το πεδίο μπορεί πλέον να αναγνωριστεί από το όνομά του slot στο οποίο αντιστοιχεί. Εκτελέστε τις παρακάτω εντολές τροποποίησης στα γεγονότα που υπάρχουν στην λίστα γεγονότων από την προηγούμενη άσκηση. Έχετε ανοιχτό το παράθυρο Facts:

1. (facts)
2. (modify 0 (name Sissy))
3. (facts)
4. (modify 5 (name Tom) (colour white))
5. (facts)
6. (modify 1 (location Athens) (address “”))
7. (facts)
8. (modify 7 (facilities sauna))
9. (facts)

10. (modify 2 (surname Papadakh))
11. (facts)

Όπως παρατηρείτε, η εντολή modify (με μια κίνηση) διαγράφει ένα μη διατεταγμένο γεγονός και εισάγει ταυτόχρονα ένα νέο στη θέση του, προσδιορίζοντας ένα ή περισσότερα slots του γεγονότος τα οποία θα αλλάξουν. Παρατηρήστε πως το γεγονός πριν και το γεγονός μετά την τροποποίηση έχουν διαφορετικούς fact-identifiers.

- Μια ακόμη εντολή, εκτός από την modify, που μπορούμε να χρησιμοποιήσουμε σε μη διατεταγμένα γεγονότα είναι η εντολή **duplicate**. Με αυτήν την εντολή δημιουργούμε ένα πανομοιότυπο αντίγραφο κάποιου γεγονότος και μετά μπορούμε να αλλάξουμε κάποιο slot του. Για να διαπιστώσετε την διαφορά από το modify εκτελέστε τα παρακάτω με ανοιχτό το παράθυρο Facts:

1. (duplicate 6 (name Sissy) (colour black))
2. (facts)
3. (fact-slot-value 6 name)
4. (fact-slot-value 6 black)
5. (fact-slot-value 6 colour)
6. (fact-slot-names 6)

7. (duplicate 6 (colour brown))
8. (facts)

9. (duplicate 8 (location Patras) (address "Ag.Andreou 16"))
10. (facts)

11. (duplicate 9 (home Athens) (height 1.70) (profession secretary))
12. (facts)
13. (fact-slot-value 9 height)
14. (fact-slot-value 9 name)
15. (fact-slot-names 9)

16. (duplicate 9 (home Patras))
17. (facts)

Τι ορίσματα παίρνουν οι εντολές **fact-slot-value** και **fact-slot-names** και τι κάνουν; Ποια είναι η διαφορά των εντολών modify και duplicate; Παρατηρήστε ότι πλεονέκτημα των δύο αυτών εντολών είναι πως δεν χρειάζεται να προσδιορίζονται μαζί με τα πεδία που αλλάζουν τιμή και τα πεδία που δεν αλλάζουν τιμή.

- Δοκιμάστε τώρα να ορίσετε εκ νέου στο CLIPS το πρότυπο cat ως εξής:

```
(deftemplate cat
  (slot name (type SYMBOL) (default ?NONE))
  (slot colour (type SYMBOL) (default unknown))
  (slot home (type SYMBOL)))
```

Τι παρατηρείτε; Στη συνέχεια ορίστε ένα καινούργιο πρότυπο ως εξής:

```
(deftemplate dog
  (slot name (type SYMBOL) (default ?NONE))
  (slot colour (type SYMBOL) (default unknown)))
```

Εκτελέστε την εντολή (**list-deftemplates**) για να δείτε αν προστέθηκε το νέο πρότυπο που ορίσατε. Χωρίς να έχουμε εισάγει κάποιο γεγονός αυτού του προτύπου ακόμα (χωρίς δηλαδή να έχουμε χρησιμοποιήσει το πρότυπο) αποφασίζουμε πως θέλουμε να ορίσουμε ξανά το πρότυπο ως εξής:

```
(deftemplate dog
  (slot name (type SYMBOL) (default ?NONE))
  (slot colour (type SYMBOL) (default unknown))
  (slot master (type SYMBOL)))
```

Εκτελέστε (**ppdeftemplate dog**). Τι συνέβη τώρα; Συνεπώς, παρατηρούμε πως ο επανακαθορισμός ενός προτύπου οδηγεί σε διαγραφή προηγούμενου ορισμού του ίδιου προτύπου. Επιπλέον, ένα πρότυπο δεν μπορεί να οριστεί ξανά από την στιγμή που έχει χρησιμοποιηθεί έστω μία φορά (πχ εισαγωγή γεγονότος του προτύπου στην λίστα γεγονότων κλπ).

➤ Εκτελώντας τις παρακάτω εντολές προσπαθήστε να καταλάβετε τι κάνουν:

1. (deftemplate-slot-allowed-values hotel facilities)
2. (deftemplate-slot-singlep hotel facilities)
3. (deftemplate-slot-singlep hotel name)
4. (deftemplate-slot-singlep family parents)
5. (deftemplate-slot-cardinality family parents)
6. (deftemplate-slot-cardinality family kids)
7. (deftemplate-slot-default-value person height)
8. (deftemplate-slot-default-value person name)
9. (deftemplate-slot-defaultp person height)
10. (deftemplate-slot-defaultp person name)
11. (deftemplate-slot-defaultp cat colour)
12. (deftemplate-slot-existp cat name)
13. (deftemplate-slot-existp hotel home)
14. (deftemplate-slot-existp hotel location)
15. (deftemplate-slot-names hotel)
16. (deftemplate-slot-names cat)
17. (deftemplate-slot-names dog)
18. (deftemplate-slot-names person)
19. (deftemplate-slot-names family)
20. (undeftemplate dog)
21. (list-deftemplates)

- 22. (undeftemplate cat)
- 23. (list-deftemplates)
- 24. (undeftemplate *)

- 25. (retract *)
- 26. (undeftemplate *)
- 27. (list-deftemplates)
- 28. (clear)

Όπως βλέπετε στις εντολές 22-24 όταν ένα πρότυπο έχει χρησιμοποιηθεί (πχ από κάποιο γεγονός ή κάποιο κανόνα) η διαγραφή του με την εντολή **undeftemplate** θα αποτύχει. Διαφορετικά το πρότυπο διαγράφεται επιτυχώς και για να μπορέσει να χρησιμοποιηθεί πρέπει να οριστεί ξανά.

- Γράψτε ορισμούς για τα παρακάτω πρότυπα γεγονότων σε ένα αρχείο που θα δημιουργήσετε, θα σώσετε και θα φορτώσετε στο CLIPS:

Πρότυπο 1: Τραγούδι με τίτλο τύπου string, συνθέτη τύπου string, στιχουργό τύπου string, ερμηνευτή τύπου string, διάρκεια τύπου float, είδος τύπου symbol με επιτρεπόμενες τιμές: rock, λαϊκό, κλασσικό, jazz, country. Το μόνο πεδίο που πρέπει να έχει υποχρεωτικά τιμή είναι ο τίτλος.

Πρότυπο 2: Μαθητής με όνομα τύπου string, επώνυμο τύπου string, ηλικία τύπου ακεραίου με τιμές από 7 έως 18, τόπο κατοικίας τύπου symbol, hobbies τύπου symbol με πλήθος από 2 έως 4. Κάθε μαθητής έχει οπωσδήποτε όνομα, επώνυμο και ηλικία, η default τιμή του τόπου κατοικίας εισάγεται από το CLIPS.

Στη συνέχεια δημιουργήστε μερικά γεγονότα που βασίζονται σε αυτά τα πρότυπα και κάντε τα assert στην λίστα γεγονότων.

- Εκτελέστε τις παρακάτω εντολές στο CLIPS μετά από τις εντολές (clear) και (clear-window):

- 1. (setgen 1)
- 2. (gensym)
- 3. (gensym)
- 4. (gensym)

- 5. (clear)
- 6. (gensym)
- 7. (gensym)

- 8. (setgen 2)
- 9. (gensym)
- 10. (gensym)

- 11. (random)
- 12. (random)
- 13. (random)
- 14. (random 20 30)

15. (random 20 30)
16. (random 30 35)
17. (random -30 -20)

Η συνάρτηση **gensym** επιστέφει ένα ειδικό σύμβολο που μπορεί να αποθηκευτεί σε ένα πεδίο. Χρησιμοποιείται κυρίως στο να βάζει ετικέτα σε πρότυπα που χρειάζονται κάποιον μοναδικό προσδιοριστή, όταν δεν μας ενδιαφέρει τι θα είναι αυτός ο προσδιοριστής. Η gensym γίνεται reset μετά από κάποια κλήση clear; Πώς μπορούμε να το κάνουμε reset ή να καθορίσουμε οι ίδιοι τον αρχικό αριθμό που θα χρησιμοποιήσει η gensym; Τι κάνει η συνάρτηση **random**;

- Αφού καθαρίσετε το περιβάλλον του CLIPS, ορίστε το παρακάτω πρότυπο:

```
(deftemplate person
  (slot name (type STRING) (default ?NONE))
  (slot personal-code (type SYMBOL) (default-dynamic (gensym))))
```

Στην συνέχεια κάνετε τις παρακάτω εισαγωγές γεγονότων:

1. (assert (person (name "Alice Johnson")))
2. (assert (person (name "Peter Johnson")))
3. (assert (person (name "Jenny Jackson")))
4. (assert (person (name "Peter Jackson")))
5. (assert (person (name "Peter Jackson")))

Δώστε την εντολή (facts). Ποια είναι η τιμή που δίνεται εξ' ορισμού με κάθε εισαγωγή person στο πεδίο personal-code; Ποια είναι η διαφορά των περιορισμών που εισάγουν οι λέξεις κλειδιά **default** και **default-dynamic**; Στη συνέχεια εκτελέστε τις παρακάτω εντολές και δείτε τι κάνουν:

- (deftemplate-slot-defaultp person name)
- (deftemplate-slot-defaultp person personal-code)
- (deftemplate A
 (slot x)
 (slot y (type SYMBOL))
 (slot z (range 3 10)))
- (deftemplate-slot-range A x)
- (deftemplate-slot-range A y)
- (deftemplate-slot-range A z)

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