

ΑΣΚΗΣΗ 5

Πρωταρχική μέθοδος αναπαράστασης γνώσης στο CLIPS είναι οι κανόνες. Στην προηγούμενη εργαστηριακή άσκηση ασχοληθήκαμε με κανόνες δύο τύπων συνθηκών: Συνθήκες Προτύπου και Συνθήκες Διεύθυνσης Προτύπου. Σε αυτήν την εργαστηριακή άσκηση εξακολουθούμε να ασχολούμαστε με τους κανόνες γενικά, ενώ παρουσιάζονται επιπλέον και **κανόνες με Εκτιμήσιμες Συνθήκες**.

Κανόνες στο CLIPS

- Δημιουργήστε ένα αρχείο .clp και μέσα σε αυτό ορίστε τους παρακάτω κανόνες και πρότυπα γεγονότων:

```
(deftemplate person
  (slot name (type SYMBOL) (default ?NONE))
  (slot surname (type SYMBOL) (default ?NONE))
  (slot sex (type SYMBOL) (allowed-symbols male female))
  (slot age (type INTEGER) (default ?NONE))
  (slot married (type SYMBOL) (allowed-symbols true false) (default false))
  (slot profession (type SYMBOL))
  (slot home (type SYMBOL)))

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

(defrule marry "a person marries another person"
  ?m <- (to-be-married ?woman ?man ?home)
  ?p1 <- (person (name ?woman) (surname ?surname1) (married false))
  ?p2 <- (person (name ?man) (surname ?surname2) (married false))
  =>
  (bind ?p1 (modify ?p1 (surname ?surname2) (married true)))
  (bind ?p2 (modify ?p2 (married true)))
  (retract ?m)
  (assert (married ?woman ?man))
  (assert (new-home ?p1 ?home))
  (assert (new-home ?p2 ?home)))

(defrule move "a person moves in a new home"
  ;(declare (salience 1))
  ?h <- (new-home ?person ?home)
  =>
  (modify ?person (home ?home))
  (retract ?h))
```

```

(defrule give-birth-first "a woman gives birth for the first time"
  ?b <- (gives-birth ?name ?baby-name ?baby-sex)
  ?m <- (married ?name ?name2)
  ?p1 <- (person (name ?name) (surname ?surname) (sex female)
          (married true) (home ?home))
  ?p2 <- (person (name ?name2))
  =>
  (bind ?p (assert (person (name ?baby-name) (surname ?surname)
                          (sex ?baby-sex) (age 0) (home ?home))))
  (assert (family (parents ?p1 ?p2) (kids (create$ ?p))))
  (retract ?b))

(defrule give-birth "a woman gives birth"
  ?b <- (gives-birth ?name ?baby-name ?baby-sex)
  ?m <- (married ?name ?name2)
  ?p1 <- (person (name ?name) (surname ?surname) (sex female)
          (married true) (home ?home))
  ?p2 <- (person (name ?name2))
  ?f <- (family (parents ?p1 ?p2) (kids $?kids))
  =>
  (bind ?p (assert (person (name ?baby-name) (surname ?surname)
                          (sex ?baby-sex) (age 0) (home ?home))))
  (modify ?f (kids (insert$ ?kids 1 ?p)))
  (retract ?b))

```

Τι σημαίνει το ';' και τι αποτέλεσμα έχει; Προσπαθήστε να εξηγήσετε τι κάνει καθένας από τους παραπάνω κανόνες. Στην συνέχεια εισάγετε τα παρακάτω γεγονότα παρατηρώντας τα παράθυρα Facts και Agenda:

- (person (name Maria) (surname Papadopoulou) (sex female) (age 28) (profession teacher) (home Patras))
- (person (name Vaggelis) (surname Anagnwstoy) (sex male) (age 35) (profession engineer) (home Athens))
- (to-be-married Maria Vaggelis Athens)

Ποιοι κανόνες ενεργοποιούνται; Δώστε την εντολή (run) και παρατηρήστε τα αποτελέσματα. Ο κανόνας marry, όταν εκτελείται, έχει επίδραση πάνω στο slot home των ατόμων που αφορά; Τι παρατηρείτε στα slot home των ατόμων που παντρεύτηκαν; Γιατί έχουν αλλάξει; Ποιοι κανόνες εκτελέστηκαν;

- Τώρα για να μην χρειάζεται να γράφετε συνέχεια τα ίδια γεγονότα κατά την εισαγωγή, ορίστε μέσα στο αρχείο με τους κανόνες και τα πρότυπα γεγονότων μια δομή deffacts για τα δύο παραπάνω μη διατεταγμένα γεγονότα. Δώστε την εντολή (clear) και φορτώστε ξανά το αρχείο σας στο CLIPS. Ποια εντολή πρέπει να δώσετε για να φορτωθούν τα γεγονότα της δομής deffacts στην λίστα γεγονότων; Δώστε αυτήν την εντολή και στην συνέχεια εισάγετε το γεγονός (to-be-married Maria Vaggelis Athens). Τι κάνουμε όταν θέλουμε η εκτέλεση κανόνων να προχωρήσει μόνο κατά ένα βήμα (δηλαδή όταν θέλουμε να εκτελεστεί μόνο ο επόμενος ενεργοποιημένος κανόνας και να σταματήσει η εκτέλεση); Χρησιμοποιήστε την εντολή αυτή προκειμένου να παρατηρήσετε σταδιακά την προηγούμενη εκτέλεση (στο παράθυρο Facts) και να δείτε τις

ενεργοποιήσεις άλλων κανόνων που συμβαίνουν κατά την διάρκεια της (Agenda). Τι θα γινόταν αν, αντί για το γεγονός (to-be-married Maria Vaggelis Athens), κάναμε assert το γεγονός (to-be-married Vaggelis Maria Athens); Δώστε την εντολή (reset) και μετά επιβεβαιώστε την απάντησή σας.

- Δώστε την εντολή (reset) και μετά την εντολή (rules) για να δείτε τους κανόνες σας. Στην συνέχεια δώστε την εντολή (clear) και έπειτα ξανά την εντολή (rules). Τι παρατηρείτε; Ποια η διαφορά της εντολής reset από την εντολή clear όσον αφορά τους κανόνες που ορίζουμε στο CLIPS; Θυμηθείτε τα τρία βασικά πράγματα που κάνει η εντολή reset όσον αφορά τα γεγονότα. Ένα από αυτά είναι η εισαγωγή του γεγονότος (initial-fact) που χρησιμεύει στο να ξεκινήσει η εκτέλεση ενός προγράμματος CLIPS, η οποία είναι αδύνατη αν δεν υπάρχουν κανόνες των οποίων οι συνθήκες να ικανοποιούνται από γεγονότα. Σημειώστε πως **ένας κανόνας χωρίς συνθήκες είναι ισοδύναμος με έναν κανόνα που έχει μοναδική συνθήκη το (initial-fact)**.
- Αλλάξτε τον κανόνα marry, έτσι ώστε οι ενέργειες:

```
(bind ?p1 (modify ?p1 (surname ?surname2) (married true)))
(bind ?p2 (modify ?p2 (married true)))
```

να αντικατασταθούν από τις ενέργειες:

```
(modify ?p1 (surname ?surname2) (married true))
(modify ?p2 (married true))
```

Στη συνέχεια, φορτώστε ξανά το αρχείο σας στο CLIPS, δώστε την εντολή reset και εισάγετε το γεγονός (to-be-married Maria Vaggelis Athens). Στην συνέχεια δώστε την εντολή (run). Μπορείτε να εξηγήσετε τι συνέβη; Σαν βοήθεια κάντε τα εξής:

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (run 1)

Τι παρατηρείτε σχετικά με τους fact identifiers στα πεδία των γεγονότων new-home που εισάγει τώρα ο κανόνας marry; Θυμηθείτε πως μια εντολή modify αφαιρεί ένα γεγονός πριν εισάγει την νέα τροποποιημένη μορφή του. Συνεπώς το τροποποιημένο γεγονός έχει διαφορετικό fact identifier από την προηγούμενη μορφή του. Διορθώστε τις αλλαγές που κάνατε στον κανόνα marry ώστε να επανέλθει στην αρχική του εκδοχή.

- Παρατηρήστε τους κανόνες give-birth-first και give-birth. Ποια είναι τα δύο σημεία στα οποία διαφέρουν μεταξύ τους; Σε ένα υποθετικό γεγονός family με το slot: (kids <Fact-18> <Fact-16> <Fact-15>), ποιος fact identifier αναφέρεται στο πιο πρόσφατα γεννημένο παιδί; Φορτώστε ξανά το αρχείο σας στο CLIPS (αφού πρώτα δώσετε την εντολή clear) και κάνετε τα παρακάτω παρατηρώντας τα αποτελέσματα:

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (run)

- (assert (gives-birth Maria Eleni female))
- (agenda)
- (run)
- (facts)

- (assert (gives-birth Maria Alexandros male))
- (agenda)
- (run)
- (facts)

- (assert (gives-birth Maria Margarita female))
- (agenda)
- (run)
- (facts)

Παρατηρήστε πως όταν ενεργοποιούνται και οι δύο κανόνες give-birth-first και give-birth είναι τυχαία η επιλογή του κανόνα που θα εκτελεστεί πρώτος.

- Αλλάξτε κατάλληλα την ενέργεια (modify ?f (kids (insert\$?kids 1 ?p))) του κανόνα give-birth, έτσι ώστε κάθε φορά να εισάγεται σαν τελευταίο πεδίο του multislot kids το πιο πρόσφατα γεννημένο παιδί. Επαναλάβετε τις προηγούμενες ενέργειες και ελέγξτε αν με την αλλαγή που κάνατε συμβαίνει αυτό που ζητείται. Παρατηρείτε κάποια συνέπεια της τυχαίας σειράς στην εκτέλεση των κανόνων give-birth-first και give-birth;
- Δώστε την εντολή reset. Κάνετε τα παρακάτω παρατηρώντας τα αποτελέσματα:
 - (set-break move)
 - (assert (to-be-married Maria Vaggelis Athens))
 - (run)

Τι παρατηρείτε; Δώστε ξανά την εντολή (run) όσες φορές χρειάζεται μέχρι να μην υπάρχει άλλος κανόνας προς εκτέλεση. Τι κάνει η εντολή **set-break**; Αφαιρέστε το break με την εντολή (**remove-break move**).

- Τώρα κάνετε τις παρακάτω ενέργειες και παρατηρήστε το αποτέλεσμα:
 - (reset)
 - (assert (to-be-married Maria Vaggelis Athens))
 - (assert (gives-birth Maria Eleni female))
 - (run)

Παρατηρήστε το slot home της Ελένης. Γιατί έχει αυτήν την τιμή;

- Όταν ικανοποιούνται οι συνθήκες διαφορετικών κανόνων ταυτόχρονα και δεν έχει καθοριστεί κάποια στρατηγική επίλυσης συγκρούσεων, ούτε έχουν καθοριστεί προτεραιότητες στους κανόνες, η επιλογή του κανόνα που θα εκτελεστεί είναι τυχαία, όπως είδαμε και πριν. Στο σημείο όπου ικανοποιούνται οι κανόνες move και give-birth ταυτόχρονα, δεν υπάρχει κάτι που θα αναγκάσει τον κανόνα move να εκτελεστεί πριν από τον give-birth. Με στρατηγικές επίλυσης συγκρούσεων και ορισμό προτεραιοτήτων θα

ασχοληθούμε εκτενέστερα σε επόμενη εργαστηριακή άσκηση, αλλά ενδεικτικά δείτε τι συμβαίνει αν αφαιρεθεί το ‘;’ από την εντολή (declare (salience 1)) του κανόνα move.

Βγάζοντας την δήλωση αυτή από τα σχόλια, ο κανόνας move αποκτά προτεραιότητα ίση με 1 (μεγαλύτερη από την προτεραιότητα 0 που εξ’ ορισμού δίνεται από το CLIPS στους υπόλοιπους κανόνες). Δώστε την εντολή clear, φορτώστε το αρχείο σας στο CLIPS μετά την αλλαγή του κανόνα move και δώστε ξανά τις παρακάτω εντολές παρατηρώντας τα παράθυρα Facts και Agenda:

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (assert (gives-birth Maria Eleni female))
- (run)

Τώρα τι τιμή έχει το slot home της Ελένης; Αν θέλετε επαναλαμβάνετε τις εντολές εκτελώντας βηματικά τους κανόνες για να παρατηρήσετε καλύτερα την διαδικασία της εκτέλεσης.

- Αντί να δώσουμε προτεραιότητα στον κανόνα move θα μπορούσαμε να ενσωματώσουμε την λειτουργία του στον κανόνα marry. Αφήστε τον κανόνα move ως έχει (θα μπορεί να χρησιμοποιηθεί για μεμονωμένες μετακομίσεις ατόμων ακόμα) αλλά αλλάξτε τον κανόνα marry έτσι ώστε να ενημερώνει κατάλληλα και τα slots home των ατόμων που παντρεύονται και να μην εισάγει τα γεγονότα new-home που εισήγαγε πριν. Καθαρίστε το περιβάλλον CLIPS, φορτώστε το αλλαγμένο αρχείο σας και δώστε ξανά τις εντολές:

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (assert (gives-birth Maria Eleni female))
- (run)

Παρατηρήστε πως το slot home της Ελένης ενημερώνεται σωστά σε αυτήν την περίπτωση, αφού και τα slots home των γονιών της έχουν ενημερωθεί σωστά πριν εκτελεστεί ο κανόνας give-birth που έχει σαν αποτέλεσμα την γέννησή της.

- **Οι βασικές ενέργειες που μπορεί να κάνει ένας κανόνας**, σύμφωνα με όσα έχουμε δει ως τώρα, είναι η **εισαγωγή**, η **διαγραφή** και η **τροποποίηση γεγονότων** (αποτέλεσμα ενός κανόνα μπορεί να είναι και η κλήση κάποιας συνάρτησης αλλά αυτό θα μας απασχολήσει σε επόμενη εργαστηριακή άσκηση). Είναι προφανές, λοιπόν, πως ένας κανόνας μπορεί με τις ενέργειές του να έχει σαν αποτέλεσμα την ενεργοποίηση ή την μη ενεργοποίηση άλλων κανόνων. Ένα τέτοιο παράδειγμα είδαμε πριν με τον κανόνα marry που ενεργοποιούσε τον κανόνα move. Στο αρχείο σας προσθέστε τους επόμενους ορισμούς:

```
(deftemplate divorced-person
  (slot name (type SYMBOL) (default ?NONE))
  (slot surname (type SYMBOL) (default ?NONE))
  (slot sex (type SYMBOL) (allowed-symbols male female))
  (slot age (type INTEGER) (default ?NONE))
  (slot profession (type SYMBOL))
  (slot home (type SYMBOL))
  (multislot kids (cardinality 1 15)))
```

```
(defrule divorce "a person divorces another person"
  ?m <- (to-be-divorced ?woman ?surname ?man ?home1 ?home2)
  ?h <- (married ?woman ?man)
  ?p1 <- (person (name ?woman) (profession ?profession1) (married true)
         (age ?age1))
  ?p2 <- (person (name ?man) (profession ?profession2) (surname ?surname1)
         (married true) (age ?age2))
  ?f <- (family (parents ?p1 ?p2) (kids $?kids))
  =>
  (assert (divorced-person (name ?woman) (surname ?surname) (sex female)
                          (age ?age1) (profession ?profession1) (home ?home1) (kids ?kids)))
  (assert (divorced-person (name ?man) (surname ?surname1) (sex male)
                          (age ?age2) (profession ?profession2) (home ?home2) (kids ?kids)))
  (assert (divorced ?woman ?man))
  (retract ?f)
  (retract ?h)
  (retract ?p1)
  (retract ?p2)
  (retract ?m))
```

Παρατηρήστε τον κανόνα `divorce` για να καταλάβετε τι κάνει. Δοκιμάστε στη συνέχεια να κάνετε τις επόμενες ενέργειες παρατηρώντας τα αποτελέσματα στα παράθυρα `Facts` και `Agenda`. Εννοείται πια πως πρώτα καθαρίζετε το `CLIPS` και φορτώνετε ξανά το αρχείο με τις αλλαγές.

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (assert (gives-birth Maria Eleni female))
- (run)
- (assert (to-be-divorced Maria Papadopoulou Vaggelis Patras Athens))
- (run)
- (assert (gives-birth Maria Margarita female))

Ποιος κανόνας ενεργοποιείται μετά την τελευταία εισαγωγή; Τι κάνει ο κανόνας `divorce` και αποτρέπει την ενεργοποίηση του κανόνα `give-birth-first` (ή `give-birth`);

- Μετά από όλα αυτά ήρθε η στιγμή να προσπαθήσετε να δημιουργήσετε οι ίδιοι έναν απλό κανόνα. Ορίστε στο αρχείο σας έναν κανόνα με όνομα `birthday` που ενεργοποιείται στα γενέθλια ενός ατόμου και αυξάνει κατά ένα την ηλικία του. Το ότι ένα άτομο με όνομα `?name` έχει γενέθλια δηλώνεται με την εισαγωγή του γεγονότος (`birthday ?name`) στην λίστα γεγονότων. Φροντίστε μετά από κάθε πιθανή εκτέλεση του κανόνα `birthday` να έχει διαγραφεί από την λίστα γεγονότων το γεγονός `birthday` που τον ενεργοποίησε. Αφού φορτώσετε το αρχείο με τον νέο σας κανόνα, εκτελέστε τις παρακάτω ενέργειες και επιβεβαιώστε πως δουλεύει σωστά.

- (reset)
- (assert (birthday Maria))

- (assert (birthday Vaggelis))
- (agenda)
- (run)
- (facts)

- (assert (birthday Maria))
- (assert (birthday Vaggelis))
- (agenda)
- (run)
- (facts)

Κανόνες με Εκτιμήσιμες Συνθήκες

- Έστω πως θέλουμε να αλλάξουμε τον κανόνα marry, έτσι ώστε να συγκρίνει τις ηλικίες των ατόμων που πρόκειται να παντρευτούν και να επιτρέπει τον γάμο μόνο αν αυτές είναι μεγαλύτερες από 20 και οι δύο και διαφέρουν μεταξύ τους κατά λιγότερο από 25 χρόνια. Διαφορετικά να μην κάνει τίποτα. Αλλάξτε τον κανόνα όπως φαίνεται παρακάτω:

```
(defrule marry "a person marries another person"
  ?m <- (to-be-married ?woman ?man ?home)
  ?p1 <- (person (name ?woman) (surname ?surname1) (married false)
          (age ?age1))
  ?p2 <- (person (name ?man) (surname ?surname2) (married false)
          (age ?age2))
  (> ?age1 20)
  (> ?age2 20)
  (< (abs (- ?age1 ?age2)) 25)
  =>
  (modify ?p1 (surname ?surname2) (married true) (home ?home))
  (modify ?p2 (married true) (home ?home))
  (retract ?m)
  (assert (married ?woman ?man)))
```

Δοκιμάστε να καθαρίσετε το CLIPS και να φορτώσετε το αρχείο με τον νέο κανόνα marry. Τι συμβαίνει; Είπαμε πως στο αριστερό μέρος ενός κανόνα πρέπει να υπάρχουν μόνο συνθήκες ικανές να ταιριάζουν-ενοποιηθούν με γεγονότα της λίστας γεγονότων και όχι εντολές που εκτελούν κάποιον υπολογισμό ή σύγκριση. Ποια από αυτά που εμφανίζονται στο δεξιό μέρος του παραπάνω κανόνα δεν αποτελούν συνθήκες αλλά εντολές;

- Υπάρχει ένας ειδικός τύπος συνθηκών που μας επιτρέπει να κάνουμε συγκρίσεις στο δεξιό μέρος ενός κανόνα. Οι συνθήκες αυτές ονομάζονται **εκτιμήσιμες** και έχουν την μορφή: **(test <function-call>)**. Μια εκτιμήσιμη συνθήκη ικανοποιείται όταν η κλήση συνάρτησης μέσα στο test επιστρέψει μια μη-FALSE τιμή. Αντικαταστήστε στον κανόνα marry τις εντολές

```
(> ?age1 20)
(> ?age2 20)
(< (abs (- ?age1 ?age2)) 25)
```

με τις παρακάτω εκτιμήσιμες συνθήκες:

```
(test (> ?age1 20))
(test (> ?age2 20))
(test (< (abs (- ?age1 ?age2)) 25))
```

Αλλάξτε κατάλληλα τα γεγονότα στη δομή deffacts του αρχείου σας ώστε η ηλικία της Μαρίας να είναι 20 και η ηλικία του Βαγγέλη 21. Μετά την διαδικασία clear και φόρτωσης του αρχείου στο CLIPS κάνετε τα παρακάτω και παρατηρήστε τα αποτελέσματα:

- (reset)
- (assert (to-be-married Maria Vaggelis Athens))
- (agenda)

Ενεργοποιείται ο κανόνας marry; Αν όχι, γιατί συμβαίνει αυτό; Στην συνέχεια αλλάξτε ξανά τα γεγονότα στη δομή deffacts ώστε η ηλικία της Μαρίας να είναι 25 και η ηλικία του Βαγγέλη 60. Μετά την διαδικασία clear και φόρτωσης του αρχείου στο CLIPS επαναλάβετε την προηγούμενη διαδικασία. Ενεργοποιήθηκε αυτή τη φορά ο κανόνας marry; Αν όχι, γιατί;

Αλλάξτε και πάλι τα γεγονότα στη δομή deffacts ώστε η ηλικία της Μαρίας να είναι 28 και η ηλικία του Βαγγέλη 35 και κάνετε ξανά τις προηγούμενες ενέργειες. Ενεργοποιείται τώρα ο κανόνας marry; Γιατί;

- Δώστε την εντολή (clear). Ορίστε στο CLIPS έναν κανόνα check-temperature ο οποίος με βάση μια θερμοκρασία ?t που δηλώνεται με το γεγονός (temperature ?t), ελέγχει αν η θερμοκρασία αυτή είναι φυσιολογική και αν ναι εισάγει το γεγονός (comfortable-temperature ?t) στη λίστα γεγονότων. Η θερμοκρασία θεωρείται φυσιολογική αν είναι μεγαλύτερη ή ίση με 20 βαθμούς Κελσίου και αν είναι μικρότερη από 30 βαθμούς Κελσίου. Ο κανόνας όταν εκτελείται διαγράφει από την λίστα γεγονότων το γεγονός που τον ενεργοποίησε. Χρησιμοποιήστε εκτιμήσιμες συνθήκες. Αφού ορίσετε τον κανόνα, εισάγετε τα επόμενα γεγονότα:

- (temperature 19)
- (temperature 20)
- (temperature 22)
- (temperature 25)
- (temperature 27)
- (temperature 30)
- (temperature 35)

Δεδομένου πως δουλεύει σωστά ο κανόνας που φτιάξατε, ποια περιμένετε να είναι τα περιεχόμενα της λίστας γεγονότων όταν δώσετε την εντολή (run); Δώστε την εντολή εκτέλεσης και ελέγξτε την σωστή λειτουργία του κανόνα σας.

- Εκτός από τους αριθμητικούς τελεστές <, >, <=, >=, = και <> (διάφορο από), στις εκτιμήσιμες συνθήκες μπορούν να χρησιμοποιηθούν και οι τελεστές σύγκρισης συμβόλων και strings: **eq** και **neq**.

Καθαρίστε ξανά το CLIPS και ορίστε σε ένα αρχείο τον παρακάτω κανόνα check-invitations ο οποίος με βάση το ονοματεπώνυμο κάποιου και την πληροφορία για το ποιος είναι καλεσμένος κάποιου, επιτρέπει ή όχι σε κάποιον την είσοδο στο συγκεκριμένο μέρος. Γεγονότα του τύπου (invited ?name ?surname) εκφράζουν την πληροφορία πως το άτομο με όνομα ?name και επώνυμο ?surname είναι καλεσμένο, ενώ γεγονότα του τύπου (wants-to-pass ?name ?surname) εκφράζουν την πληροφορία πως το άτομο με όνομα ?name και επώνυμο ?surname θέλει να περάσει:

```
(defrule check-invitation
  ?p <- (wants-to-pass ?name ?surname)
  (invited ?name ?surname)
=>
  (assert (passes ?name ?surname))
  (retract ?p))
```

Στην συνέχεια ορίστε στο ίδιο αρχείο την παρακάτω δομή deffacts με τους καλεσμένους:

```
(deffacts invited
  (invited Maria Spyropoulou)
  (invited Athina Spyropoulou)
  (invited Vasilhs Papathanasioy)
  (invited Antonis Aleksiadis)
  (invited Vaggelis Anagnwstoy)
  (invited Roula Papadakh))
```

Φορτώστε το αρχείο σας και κάνετε τις εξής ενέργειες παρατηρώντας τα παράθυρα Facts και Agenda:

- (reset)
- (assert (wants-to-pass Maria Papadopoulou))
- (assert (wants-to-pass Maria Spyropoulou))
- (assert (wants-to-pass Aleksis Aleksiou))
- (assert (wants-to-pass Athina Spyropoulou))
- (assert (wants-to-pass Vaggelis Anagnwstoy))
- (assert (wants-to-pass Roula Papadakh))
- (run)
- (facts)

Αλλάξτε τον κανόνα check-invitations ώστε να κάνει αυτό που έκανε και πριν, χρησιμοποιώντας όμως τώρα **περιορισμούς κατηγορήματος** τους οποίους συναντήσαμε στην προηγούμενη εργαστηριακή άσκηση. Φορτώστε ξανά το αλλαγμένο αρχείο και επαναλάβετε την προηγούμενη διαδικασία για να δείτε αν ο κανόνας μετά τις αλλαγές σας δουλεύει όπως πριν από αυτές.

Αλλάξτε ξανά τον κανόνα check-invitations ώστε να κάνει αυτό που έκανε και πριν, χρησιμοποιώντας όμως τώρα **εκτιμήσιμες συνθήκες**. Φορτώστε ξανά το αλλαγμένο αρχείο και επαναλάβετε την προηγούμενη διαδικασία για να δείτε αν ο κανόνας μετά τις αλλαγές σας δουλεύει όπως πριν από αυτές.

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