

ΤΕΛΙΚΗ ΚΑΤΑΘΕΣΗ ΙΑΝΟΥΑΡΙΟΣ 19, 2020

ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ PARKING

ΕΡΓΑΣΙΑ 2: ΈΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ (CLIPS)

ΜΑΘΗΜΑ:
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (Ε) 2020/21

ΒΕΛΑΣΚΟ ΠΑΟΛΑ cs161020
ΜΙΧΑ ΕΥΑΓΓΕΛΙΑ cs171102

(Καθηγητές: Γεωργούλη Κατερίνα,
Βουλόδημος Θάνος, Τσελέντη Παναγιώτα)

ΤΜΗΜΑ 3 (ΝΕΟΙ) ΤΡΙΤΗ 15:00-17:00

Πίνακας περιεχομένων

Α' ΜΕΡΟΣ – Αρχικό Πρόβλημα	4
Περιγραφή του προβλήματος	4
Το πρόβλημα του Parking	4
Μοντελοποίηση του προβλήματος.....	5
Χώρος καταστάσεων.....	5
Αρχική κατάσταση	5
Στόχος – Τελική Κατάσταση	5
Κανόνες	5
Αναπαράσταση του Κόσμου του Προβλήματος	7
Γεγονότα που περιγράφουν τα δεδομένα του προβλήματος.....	7
Παρουσίαση των κανόνων	9
I. begin.....	9
II. check_capacity	9
III. move_platform_east.....	10
IV. move_platform_north	11
V. move_platform_west.....	12
VI. move_platform_south	12
VII. enter_parking.....	12
VIII. goal.....	13
Πρότυπος Κώδικας	14
Αποτελέσματα Εκτέλεσης	18
Τρία αυτοκίνητα – random	18
Τρία αυτοκίνητα – simplicity	18
Τρία αυτοκίνητα – complexity	19
Τρία αυτοκίνητα – depth	19
Τρία αυτοκίνητα – width	19
Τρία αυτοκίνητα – lex	20
Τρία αυτοκίνητα – mea.....	20
Πέντε αυτοκίνητα – random.....	20
Επτά αυτοκίνητα – depth.....	23
Β' ΜΕΡΟΣ – Επέκταση προβλήματος	24

Περιγραφή νέου προβλήματος	24
Το πρόβλημα του Parking – εκτεταμένη εκδοχή	24
Υλοποίηση προβλήματος - Απαιτούμενα	25
Μοντελοποίηση.....	26
Αρχική κατάσταση	26
Κατάσταση στόχου	26
Νέοι Κανόνες	26
Αναπαράσταση του Νέου Κόσμου Προβλήματος	27
Τροποποίηση υπάρχων κανόνων	27
I. Begin	27
II. allparked_goal.....	27
Παρουσίαση νέων κανόνων	27
III. choice	27
IV. add_car	28
V. want_exit	29
VI. exit_parking	30
VII. exit_done	30
VIII. move_platform_up_to_1.....	31
IX. move_platform_up_to_2.....	32
X. move_platform_down_from_2	32
XI. move_platform_down_from_1	32
XII. empty_parking	33
XIII. all_platforms_full.....	33
XIV. car_not_exists	33
Πρότυπος κώδικας.....	34
Αποτελέσματα Εκτέλεσης	41
Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική simplicity.....	41
Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική complexity	41
Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική depth	41
Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική breadth.....	41
Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική random	41
Εισαγωγή 5 αυτοκινήτων – Εξαγωγή 3 αυτοκινήτων: στρατηγική random	46
Εισαγωγή 17 αυτοκινήτων: στρατηγική random.....	50

Συμπεράσματα – Παρατηρήσεις	52
Παρατηρήσεις.....	52
Συμπεράσματα.....	53
Παράρτημα	54
Στρατηγική random.....	54
Στρατηγική Simplicity.....	59
Στρατηγική Complexity	62
Βιβλιογραφία	65

Α' ΜΕΡΟΣ – Αρχικό Πρόβλημα

Περιγραφή του προβλήματος

Το πρόβλημα του Parking

Στο πρόβλημα που παρουσιάζουμε, υπάρχει ένα parking που λειτουργεί με αυτόματο παρκαδόρο για την είσοδο στο χώρο και την τοποθέτηση σε πλατφόρμες παρκαρίσματος των αυτοκινήτων που είναι σε αναμονή. Το parking διαθέτει ένα επίπεδο λειτουργίας με 6 χώρους υποδοχής πλατφορμών, ο ένας από τους οποίους βρίσκεται στην πρόσοψη στη μέση και χαρακτηρίζεται ως χώρος εισόδου.

Οι χώροι στο υπάρχον επίπεδο (1 ο επίπεδο) ονομάζονται S01-S06, με χώρο εισόδου τον S02. Κάθε χώρος (SPACE) χαρακτηρίζεται από το όνομά του, από το επίπεδο στο οποίο βρίσκεται (στην επέκταση του προβλήματος θα υπάρχουν περισσότερα του ενός επίπεδα) και από την κατάστασή του, δηλαδή αν είναι ελεύθερος ή αν υπάρχει επάνω τοποθετημένη πλατφόρμα.

Οι πλατφόρμες στο επίπεδο 1 είναι 5, με ονόματα από P1-P5. Κάθε πλατφόρμα (PLATFORM) χαρακτηρίζεται από το όνομά της, την κατάστασή της (άδεια ή με παρκαρισμένο αυτοκίνητο πάνω της) και τον χώρο πάνω στον οποίο βρίσκεται. Στην αρχή λειτουργίας του parking όλες οι πλατφόρμες είναι άδειες και τοποθετημένες πάνω στους χώρους του επιπέδου 1 όπως φαίνεται στο σχήμα Α. Μια πλατφόρμα μπορεί να μετακινηθεί από το χώρο που βρίσκεται σε έναν γειτονικό χώρο αρκεί αυτός να είναι ελεύθερος. Η πλατφόρμα που μετακινείται μπορεί να είναι άδεια ή φορτωμένη με αυτοκίνητο.

Για να μπορέσει να εισέλθει ένα αυτοκίνητο (CAR) στο parking, θα πρέπει ο οδηγός του πριν το αφήσει στο χώρο αναμονής (έξω από το parking) να το έχει δηλώσει στο σύστημα παρακολούθησης, δίνοντας τον αριθμό κυκλοφορίας του. Το πλήθος των δηλωμένων αυτοκινήτων στην αρχή του προβλήματος πρέπει να είναι τουλάχιστον 1 και κάθε αυτοκίνητο χαρακτηρίζεται από τον αριθμό κυκλοφορίας του και την κατάσταση στην οποία βρίσκεται (σε αναμονή εισόδου ή παρκαρισμένο σε συγκεκριμένη πλατφόρμα). Για να μπορέσει ένα αυτοκίνητο να μπει στο parking, θα πρέπει να υπάρχει τουλάχιστον μια ελεύθερη πλατφόρμα σε οποιονδήποτε χώρο του parking η οποία να έχει μετακινηθεί από τον αυτόματο παρκαδόρο στον χώρο εισόδου.



Μοντελοποίηση του προβλήματος

Χώρος καταστάσεων

Ως χώρος καταστάσεων (state space) ορίζεται το σύνολο των έγκυρων καταστάσεων που μπορεί να βρεθεί ένα πρόβλημα κατά την εξέλιξη του κόσμου του. Έγκυρες καταστάσεις είναι η εισαγωγή αυτοκινήτου στο parking εφόσον υπάρχει ελεύθερη πλατφόρμα στο χώρο εισαγωγής καθώς και η μετακίνηση πλατφόρμας σε γειτονικό χώρο εφόσον είναι ελεύθερος.

Αρχική κατάσταση

Ο κόσμος του προβλήματος αποτελείται από 6 χώρους στάθμευσης (spaces) αριθμημένους από S01 έως S06. Ο S02 χαρακτηρίζεται ως χώρος υποδοχής. Επίσης, υπάρχουν 5 πλατφόρμες εναπόθεσης αυτοκινήτων (platforms), αριθμημένες από P01 έως P05 και είναι τοποθετημένες ανά μία σε 5 χώρους στάθμευσης.

Στην αρχική κατάσταση του προβλήματος, το parking είναι άδειο από αυτοκίνητα δηλαδή όλες οι πλατφόρμες είναι κενές και στον χώρο υποδοχής δεν υπάρχει πλατφόρμα εναπόθεσης.

Στόχος – Τελική Κατάσταση

Η τελική κατάσταση του προβλήματος είναι η επιτυχής εισαγωγή όλων των αυτοκινήτων που βρίσκονται σε αναμονή στο parking.

Κανόνες

Ως κανόνες ορίζουμε:

- I. Τη μετακίνηση των πλατφορμών με τους γείτονές τους (west, east, south , north)
- II. Την αρχική κατάσταση
- III. Τελική κατάσταση
- IV. Είσοδος αυτοκινήτου στο πάρκινγκ
- V. Έλεγχος αριθμών αυτοκινήτων

Το πρόγραμμα ξεκινάει με τον κανόνα `begin` στον οποίο προσδιορίζουμε ποια στρατηγική θα ακολουθήσει κατά την εκτέλεσή του.

Οι κανόνες `move_platform_east`, `move_platform_west`, `move_platform_north` και `move_platform_south` ορίζουν την μετακίνηση των πλατφορμών ανάμεσα στους χώρους στάθμευσης. Υπενθυμίζουμε ότι μια πλατφόρμα μπορεί να μετακινηθεί στο γειτονικό του χώρου μόνο αν ο γειτονικός αυτός χώρος δεν περιέχει κάποια πλατφόρμα.

Για την υλοποίηση τελικής κατάστασης δημιουργούμε τον κανόνα `goal`, ο οποίος ενεργοποιείται όταν πλέον δεν υπάρχουν αυτοκίνητα σε αναμονή. Με άλλα λόγια, όλα τα γεγονότα αυτοκινήτων έχουν ως κατάσταση κάποιο όνομα πλατφόρμας.

Για την είσοδο ενός αυτοκινήτου στο σύστημα του αυτόματου πάρκινγκ, πυροδοτείται ο κανόνας `enter_parking`. Για να ενεργοποιηθεί αυτός, αρκεί κάποια άδεια πλατφόρμα να βρίσκεται στο χώρο υποδοχής.

Τέλος, ορίσαμε και έναν επιπλέον απαραίτητο κανόνα, `check_capacity`, για τον έλεγχο επιτυχής εύρεσης λύσης του προβλήματος.

Αναπαράσταση του Κόσμου του Προβλήματος

Για την αναπαράσταση του προβλήματος χρησιμοποιήθηκε το εργαλείο CLIPS ώστε να προσαρμοστεί ανάλογα σε ένα Έμπειρο Σύστημα.

Γεγονότα που περιγράφουν τα δεδομένα του προβλήματος

Αρχικά, στο πρόγραμμα μας ορίζουμε τρία πρότυπα, το καθένα με τις κατάλληλες ιδιότητες όπως φαίνονται παρακάτω.

Το πρότυπο «space» έχει τρεις ιδιότητες:

- `s_name` : το όνομα του χώρου στάθμευσης
- `s_floor`: το όνομα επιπέδου στο οποίο βρίσκεται
- `s_state`: την κατάσταση που βρίσκεται, αν έχει πλατφόρμα (P) ή όχι (F)

```
(deftemplate space
  (slot s_name (type SYMBOL))
  (slot s_floor (type INTEGER))
  (slot s_state (type SYMBOL))
)
```

Το πρότυπο «platform» έχει τρεις ιδιότητες:

- `p_name` : το όνομα της πλατφόρμας
- `p_space`: το όνομα του χώρου στάθμευσης στον οποίο βρίσκεται
- `p_state`: την κατάσταση που βρίσκεται, αν έχει αυτοκίνητο ή όχι

```
(deftemplate platform
  (slot p_name (type SYMBOL))
  (slot p_space (type SYMBOL))
  (slot p_state (type SYMBOL))
)
```

Το πρότυπο «car» έχει τρεις ιδιότητες:

- `c_plate` : αριθμός πινακίδας (Μορφή πχ: 001)
- `c_state` : σε ποια πλατφόρμα βρίσκεται ή αν είναι σε κατάσταση αναμονής

```
(deftemplate car
  (slot c_plate (type INTEGER))
  (slot c_state (type SYMBOL))
)
```

Στη συνέχεια, εισάγουμε τα γεγονότα που υλοποιούνται σύμφωνα με τα αντίστοιχά τους templates.

```
(defacts spaces
  (space (s_name S01) (s_floor 1) (s_state P))
  (space (s_name S02) (s_floor 1) (s_state F))
  (space (s_name S03) (s_floor 1) (s_state P))
)
```



```
(space (s_name S04) (s_floor 1) (s_state P))  
(space (s_name S05) (s_floor 1) (s_state P))  
(space (s_name S06) (s_floor 1) (s_state P))  
)
```

```
(deffacts platforms  
  (platform (p_name P1) (p_space S01) (p_state empty))  
  (platform (p_name P2) (p_space S03) (p_state empty))  
  (platform (p_name P3) (p_space S04) (p_state empty))  
  (platform (p_name P4) (p_space S05) (p_state empty))  
  (platform (p_name P5) (p_space S06) (p_state empty))  
)
```

```
(deffacts cars  
  (car (c_plate 101) (c_state waiting))  
  (car (c_plate 102) (c_state waiting))  
  (car (c_plate 103) (c_state waiting))  
  (car (c_plate 104) (c_state waiting))  
  (car (c_plate 105) (c_state waiting))  
)
```

Επιπλέον, προσθέτουμε γεγονότα που περιγράφουν τη γειτνίαση των χώρων. Όπως παρατηρούμε παρακάτω, για παράδειγμα, ο S01 έχει στα δεξιά του, τον S02. Για τη γειτνίαση προς τα δυτικά και προς τα νότια χρησιμοποιήθηκαν τα ίδια γεγονότα αντιστρέφοντας λογικά τους χώρους στους οποίους αναφέρονται.

```
(deffacts neighbours  
  (east S01 S02)  
  (east S02 S03)  
  (east S04 S05)  
  (east S05 S06)  
  (north S01 S04)  
  (north S02 S05)  
  (north S03 S06)  
)
```

Τέλος, προσθέτουμε το γεγονός capacity για να δηλώσουμε ότι το πρόβλημά μας έχει υλοποιηθεί για 5 εισαγωγή μέχρι των πέντε αυτοκινήτων .

```
(deffacts initial  
  (capacity 5)  
)
```

Παρουσίαση των κανόνων

Έπειτα ορίζουμε τους κανόνες

1. Begin
2. Check_capacity
3. Move_platform_east
4. Move_platform_north
5. Move_platform_west
6. Move_platform_south
7. Enter_parking
8. Goal

Ένας κανόνας γίνεται ενεργός (triggered) όταν ικανοποιούνται οι συνθήκες του από τα γεγονότα της fact-list.

I. begin

Το πρόγραμμά μας ξεκινά από τον κανόνα με τη μεγαλύτερη προτεραιότητα η οποία ορίζεται από την τιμή του salience. Ορίζουμε λοιπόν το begin με salience 100, στον οποίο αρχικοποιείται και η στρατηγική που θα χρησιμοποιηθεί. Για το παράδειγμα αυτό, όταν πυροδοτηθεί ο συγκεκριμένος κανόνας, θα γίνει η χρήση της random στρατηγικής και εκτυπώνεται το μήνυμα ότι θα ξεκινήσει η εισαγωγή των αυτοκινήτων.

```
(defrule begin "Αρχικοποίηση στρατηγικής και έναρξης προγράμματος"
  (declare (salience 100))
  (initial-fact)
=>
  (set-strategy random)
  (printout t "Parking Begins !" crlf crlf)
)
```

II. check_capacity

Ο κανόνας check_capacity ενεργοποιείται μετά από την begin με salience 99, ο οποίος ελέγχει αν το πρόβλημα αρχικά μπορεί να βρει λύση, καθώς ο στόχος μας είναι να τοποθετηθούν όλα τα αυτοκίνητα σε κάποια άδεια πλατφόρμα. Αν ο αριθμός των αυτοκινήτων είναι παραπάνω από τις διαθέσιμες ελεύθερες πλατφόρμες που υπάρχουν στο πάρκινγκ, τότε είναι προφανές ότι το πρόβλημά μας δε βρίσκει λύση ποτέ.

```
(defrule check_capacity "Έλεγχος αυτοκινήτων και διαθέσιμων πλατφορμών"
  (declare (salience 99))
  ?x <- (car (c_plate ?) (c_state ?))
  (capacity ?cap)
  (test (> (length (find-all-facts ((?c car)) (eq ?c:c_state waiting) ) )
?cap ) )
=>
```

```
(printout t "More cars than spaces!" crlf)
(halt)
)
```

III. move_platform_east

Ύστερα, εκτελούνται οι κανόνες μετακίνησης, στους οποίους έχουμε ορίσει την ίδια τιμή *saliency* 50, μεταξύ τους. Αυτό οφείλεται στο γεγονός πως αν δεν έχουν την ίδια τιμή, το πρόγραμμά μας θα βρεθεί σε μια ατέρμονη επανάληψη.

Μια πλατφόρμα μπορεί να μετακινηθεί από το χώρο που βρίσκεται σε έναν γειτονικό χώρο αρκεί αυτός να είναι ελεύθερος. Ο κανόνας *move_platform_east* υλοποιεί τη μετακίνηση της πλατφόρμας προς τα δεξιά.

```
(defrule move_platform_east "Μετακίνηση πλατφόρμας δεξιά"
(declare (saliency 50))
  (east ?s ?s_right)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_right) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?)
)
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_right))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_right crlf)
)
```

Επεξήγηση κώδικα

Αποθηκεύουμε τις ταυτότητες των γεγονότων που γίνεται η ταυτοποίηση με τις συνθήκες, ώστε να μπορέσουμε έπειτα να τροποποιήσουμε αυτά τα γεγονότα με τις καινούριες καταστάσεις.

```
(east ?s ?s_right)
```

Ταυτοποίηση κάποιου γεγονότος *east* για να γίνουν οι παρακάτω έλεγχοι.

```
?x <- (space (s_name ?s) (s_floor ?) (s_state P))
```

Ελέγχει αν ο χώρος (*s_name ?s*) έχει πλατφόρμα.

```
?y <- (space (s_name ?s_right) (s_floor ?) (s_state F))
```

Ελέγχει αν ο γείτονας (*s_name ?s_right*) του (*s_name ?s*) έχει άδεια πλατφόρμα.

```
?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
```

Αποθηκεύεται η ταυτότητα του γεγονότος πλατφόρμας που υπάρχει στο χώρο που υπάρχει πλατφόρμα. Στη συνέχεια, αν ικανοποιούνται όλες οι συνθήκες ακολουθούν οι εξής αλλαγές αυτών των γεγονότων

```
(modify ?x (s_state F))
```

Αλλάζει την κατάσταση του χώρου (s_name ?s) ότι δεν υπάρχει πλατφόρμα.

```
(modify ?y (s_state P))
```

Αλλάζει την κατάσταση του χώρου (s_name ?s_right) ότι υπάρχει πλατφόρμα.

```
(modify ?z (p_space ?s_right))
```

Αλλάζει το χώρο στον οποίο βρίσκεται πλέον η πλατφόρμα (p_name ?pname).

Παράδειγμα εκτέλεσης κανόνα

Ως προς πυροδότηση κανόνα με τα παρακάτω γεγονότα

```
(east S02 S03)
(space (s_name S02) (s_floor 1) (s_state P))
(space (s_name S03) (s_floor 1) (s_state F))
(platform (p_name P2) (p_space S02) (p_state full))
```

Μετά την εκτέλεση του κανόνα

```
(space (s_name S02) (s_floor 1) (s_state F))
(space (s_name S03) (s_floor 1) (s_state P))
(platform (p_name P2) (p_space S03) (p_state empty))
```

IV. move_platform_north

Ο κανόνας move_platform_north λειτουργεί με τον παρόμοιο τρόπο όπως move_platform_east με τη διαφορά ότι η μετακίνηση της πλατφόρμας να γίνεται με τον πάνω γειτονικό του χώρο.

```
(defrule move_platform_north "Μετακίνηση πλατφόρμας πάνω"
(declare (salience 50))
(north ?s ?s_above)
?x <- (space (s_name ?s) (s_floor ?) (s_state P))
?y <- (space (s_name ?s_above) (s_floor ?) (s_state F))
?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
)
=>
(modify ?x (s_state F))
(modify ?y (s_state P))
(modify ?z (p_space ?s_above))

(printout t "platform " ?pname " moved from " ?s " to " ?s_above crlf)
)
```

V. move_platform_west

Ο κανόνας `move_platform_west` λειτουργεί με τον παρόμοιο τρόπο όπως `move_platform_east` με τη διαφορά ότι η μετακίνηση της πλατφόρμας να γίνεται με τον αριστερό γειτονικό του χώρο.

```
(defrule move_platform_west "Μετακίνηση πλατφόρμας αριστερά"
(declare (salience 50))
  (east ?s_left ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_left) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_left))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_left crlf)
)
```

VI. move_platform_south

Ο κανόνας `move_platform_south` λειτουργεί με τον παρόμοιο τρόπο όπως `move_platform_east` με τη διαφορά ότι η μετακίνηση της πλατφόρμας να γίνεται με τον από κάτω γειτονικό του χώρο.

```
(defrule move_platform_south "Μετακίνηση πλατφόρμας κάτω"
(declare (salience 50))
  (south ?s_down ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_down) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_down))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_down crlf)
)
```

VII. enter_parking

Ο κανόνας `enter_parking` ενεργοποιείται όταν στο χώρο εισόδου S02 υπάρχει πλατφόρμα η οποία είναι κενή. Αν ικανοποιείται αυτή η προϋπόθεση, τότε θα τοποθετηθεί ένα αυτοκίνητο σε αυτή

την πλατφόρμα. Επιπλέον, θα ακολουθήσουν οι κατάλληλες αλλαγές στα υπάρχον γεγονότα, ώστε να εισαχθούν οι καινούριες καταστάσεις.

```
(defrule enter_parking "Είσοδος αυτοκινήτου"
(declare (salience 98))
  ?x <- (car (c_plate ?plate) (c_state waiting))
  ?y <- (platform (p_name ?pname) (p_space S02) (p_state empty))
=>
  (modify ?x (c_state ?pname))
  (modify ?y (p_state full))

  (printout t "Car with plate: " ?plate " parked on platform " ?pname crlf)
)
```

Επεξήγηση κώδικα

- Αν υπάρχει αυτοκίνητο με κατάσταση (c_state waiting)
- Και υπάρχει και άδεια πλατφόρμα (p_name ?pname) στο χώρο S02
- Τότε να αλλάξει την κατάσταση του αυτοκινήτου (c_state ?pname) με το όνομα της πλατφόρμας στον οποίο έχει τοποθετηθεί καθώς να αλλάξει
- Και η κατάσταση της πλατφόρμας με όνομα (p_name ?pname) σε (p_state full)

VIII. goal

Ο κανόνας goal ελέγχει αν έχουν τοποθετηθεί όλα τα αυτοκίνητα που ορίσαμε στην αρχή του προβλήματός μας σε κάποια πλατφόρμα στο πάρκινγκ. Όταν ενεργοποιηθεί αυτός ο κανόνας, επισημαίνεται και η λήξη του προγράμματος. Ο κανόνας ελέγχει αν δεν υπάρχουν αυτοκίνητα στα γεγονότα με κατάσταση c_state waiting. Αν ισχύει, τότε σταματά το πρόγραμμα.

```
(defrule goal "Τερματισμός προγράμματος - Εύρεση λύσης"
(declare (salience 100))
(not (car (c_plate ?) (c_state waiting)))
=>
  (printout t "Parking is full" crlf)
  (halt)
)
```

Πρότυπος Κώδικας

```
// -----  
//                               Πρότυπα Γεγονότων  
// -----  
  
(deftemplate space  
  (slot s_name (type SYMBOL))  
  (slot s_floor (type INTEGER))  
  (slot s_state (type SYMBOL))  
)  
  
(deftemplate platform  
  (slot p_name (type SYMBOL))  
  (slot p_space (type SYMBOL))  
  (slot p_state (type SYMBOL))  
)  
  
(deftemplate car  
  (slot c_plate (type INTEGER))  
  (slot c_state (type SYMBOL))  
)  
  
// -----  
//                               Αρχικά Γεγονότα  
// -----  
  
(def facts spaces  
  (space (s_name S01) (s_floor 1) (s_state P))  
  (space (s_name S02) (s_floor 1) (s_state F))  
  (space (s_name S03) (s_floor 1) (s_state P))  
  (space (s_name S04) (s_floor 1) (s_state P))  
  (space (s_name S05) (s_floor 1) (s_state P))  
  (space (s_name S06) (s_floor 1) (s_state P))  
)  
  
(def facts platforms  
  (platform (p_name P1) (p_space S01) (p_state empty))  
  (platform (p_name P2) (p_space S03) (p_state empty))  
  (platform (p_name P3) (p_space S04) (p_state empty))  
  (platform (p_name P4) (p_space S05) (p_state empty))  
  (platform (p_name P5) (p_space S06) (p_state empty))  
)  
  
(def facts neighbours  
  (east S01 S02)  
  (east S02 S03)  
  (east S04 S05)  
  (east S05 S06)  
  (north S01 S04)  
  (north S02 S05)
```

```
(north S03 S06)
)

(deffacts cars
  (car (c_plate 101) (c_state waiting))
  (car (c_plate 102) (c_state waiting))
  (car (c_plate 103) (c_state waiting))
  (car (c_plate 104) (c_state waiting))
  (car (c_plate 105) (c_state waiting))
)

(deffacts initial
  (capacity 5)
)

;; -----
;;                                 Κανόνες
;; -----

(defrule begin "Αρχικοποίηση στρατηγικής και έναρξης προγράμματος"
  (declare (salience 100))
  (initial-fact)
=>
  ;; Ανάθεση στρατηγικής "random"
  (set-strategy random)
  ;; (set-strategy depth)
  ;; (set-strategy breadth)
  ;; (set-strategy simplicity)
  ;; (set-strategy complexity)

  (printout t "Parking Begins !" crlf crlf)
)

(defrule check_capacity "Έλεγχος αυτοκινήτων και διαθέσιμων πλατφορμών"
  (declare (salience 99))
  ?x <- (car (c_plate ?) (c_state ?))
  (capacity ?cap)
  (test (> (length (find-all-facts ((?c car)) (eq ?c:c_state waiting) ) )
?cap ) )
=>
  (printout t "More cars than spaces!" crlf)
  (halt)
)

(defrule enter_parking "Είσοδος αυτοκινήτου"
  (declare (salience 98))
  ?x <- (car (c_plate ?plate) (c_state waiting))
  ?y <- (platform (p_name ?pname) (p_space S02) (p_state empty))
=>
  (modify ?x (c_state ?pname))
  (modify ?y (p_state full))
)
```



```
(printout t "Car with plate: " ?plate " parked on platform " ?pname crlf)
)

(defrule goal "Τερματισμός προγράμματος - Εύρεση λύσης"
  (declare (salience 100))
  (not (car (c_plate ?) (c_state waiting))))
=>
  (printout t "Parking is full" crlf)
  (halt)
)

(defrule move_platform_east "Μετακίνηση πλατφόρμας δεξιά"
  (declare (salience 50))
  (east ?s ?s_right)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_right) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?)
)
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_right))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_right crlf)
)

(defrule move_platform_north "Μετακίνηση πλατφόρμας πάνω"
  (declare (salience 50))
  (north ?s ?s_above)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_above) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?)
)
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_above))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_above crlf)
)

(defrule move_platform_west "Μετακίνηση πλατφόρμας αριστερά"
  (declare (salience 50))
  (west ?s ?s_left ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_left) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
```

```
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_left))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_left crlf)
)

(defrule move_platform_south "Μετακίνηση πλατφόρμας κάτω"
(declare (salience 50))
  (north ?s_down ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_down) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_down))

  (printout t "platform " ?pname " moved from " ?s " to " ?s_down crlf)
)
```

Αποτελέσματα Εκτέλεσης

Παρακάτω θα παρατεθούν τα αποτελέσματα διάφορων εκτελέσεων χωρίς παρακολούθηση γεγονότων και κανόνων με σκοπό την γρήγορη κατανόηση των αλλαγών καθώς και την εξοικονόμηση χώρου. Οι ενδεικτικές εκτελέσεις με παρακολούθηση γεγονότων και κανόνων για κάθε παράδειγμα βρίσκονται στο παράρτημα.

Τρία αυτοκίνητα – random

```
CLIPS> (run)
Parking Begins !

platform P1 moved from S01 to S02
Car with plate: 101 parked on platform P1
platform P3 moved from S04 to S01
platform P3 moved from S01 to S04
platform P1 moved from S02 to S01
platform P1 moved from S01 to S02
platform P1 moved from S02 to S01
platform P2 moved from S03 to S02
Car with plate: 103 parked on platform P2
platform P2 moved from S02 to S03
platform P2 moved from S03 to S02
platform P2 moved from S02 to S03
platform P4 moved from S05 to S02
Car with plate: 102 parked on platform P4
Parking is full
CLIPS>
```

Τρία αυτοκίνητα – simplicity

```
CLIPS> (run)
Parking Begins !

platform P1 moved from S01 to S02
Car with plate: 103 parked on platform P1
platform P3 moved from S04 to S01
platform P4 moved from S05 to S04
platform P5 moved from S06 to S05
platform P2 moved from S03 to S06
platform P1 moved from S02 to S03
```

```
platform P5 moved from S05 to S02
Car with plate: 102 parked on platform P5
platform P2 moved from S06 to S05
platform P1 moved from S03 to S06
platform P5 moved from S02 to S03
platform P2 moved from S05 to S02
Car with plate: 101 parked on platform P2
Parking is full
CLIPS>
```

Τρία αυτοκίνητα – complexity

```
CLIPS> (run)
Parking Begins !

platform P1 moved from S01 to S02
Car with plate: 103 parked on platform P1
platform P3 moved from S04 to S01
platform P4 moved from S05 to S04
platform P5 moved from S06 to S05
platform P2 moved from S03 to S06
platform P1 moved from S02 to S03
platform P5 moved from S05 to S02
Car with plate: 102 parked on platform P5
platform P2 moved from S06 to S05
platform P1 moved from S03 to S06
platform P5 moved from S02 to S03
platform P2 moved from S05 to S02
Car with plate: 101 parked on platform P2
Parking is full
CLIPS>
```

Τρία αυτοκίνητα – depth

Ατέρμονος

Τρία αυτοκίνητα – width

Ατέρμονος

Τρία αυτοκίνητα – lex

Αιέριμος

Τρία αυτοκίνητα – mea

Αιέριμος

Πέντε αυτοκίνητα – random

CLIPS> (run)

Parking Begins !

```
platform P1 moved from S01 to S02
Car with plate: 102 parked on platform P1
platform P3 moved from S04 to S01
platform P4 moved from S05 to S04
platform P4 moved from S04 to S05
platform P3 moved from S01 to S04
platform P3 moved from S04 to S01
platform P3 moved from S01 to S04
platform P3 moved from S04 to S01
platform P4 moved from S05 to S04
platform P5 moved from S06 to S05
platform P5 moved from S05 to S06
platform P1 moved from S02 to S05
platform P1 moved from S05 to S02
platform P5 moved from S06 to S05
platform P2 moved from S03 to S06
platform P2 moved from S06 to S03
platform P2 moved from S03 to S06
platform P2 moved from S06 to S03
platform P2 moved from S03 to S06
platform P2 moved from S06 to S03
platform P2 moved from S03 to S06
platform P2 moved from S06 to S03
platform P2 moved from S03 to S06
platform P2 moved from S06 to S03
platform P2 moved from S03 to S06
platform P1 moved from S02 to S03
```

platform P3 moved from S01 to S02
Car with plate: 104 parked on platform P3
platform P3 moved from S02 to S01
platform P3 moved from S01 to S02
platform P3 moved from S02 to S01
platform P3 moved from S01 to S02
platform P3 moved from S02 to S01
platform P5 moved from S05 to S02
Car with plate: 105 parked on platform P5
platform P2 moved from S06 to S05
platform P1 moved from S03 to S06
platform P1 moved from S06 to S03
platform P1 moved from S03 to S06
platform P1 moved from S06 to S03
platform P2 moved from S05 to S06
platform P4 moved from S04 to S05
platform P3 moved from S01 to S04
platform P3 moved from S04 to S01
platform P4 moved from S05 to S04
platform P2 moved from S06 to S05
platform P1 moved from S03 to S06
platform P5 moved from S02 to S03
platform P3 moved from S01 to S02
platform P3 moved from S02 to S01
platform P3 moved from S01 to S02
platform P4 moved from S04 to S01
platform P2 moved from S05 to S04
platform P1 moved from S06 to S05
platform P5 moved from S03 to S06
platform P3 moved from S02 to S03
platform P1 moved from S05 to S02
platform P1 moved from S02 to S05
platform P1 moved from S05 to S02
platform P2 moved from S04 to S05
platform P4 moved from S01 to S04
platform P1 moved from S02 to S01
platform P2 moved from S05 to S02
Car with plate: 101 parked on platform P2

platform P2 moved from S02 to S05
platform P1 moved from S01 to S02
platform P4 moved from S04 to S01
platform P4 moved from S01 to S04
platform P1 moved from S02 to S01
platform P1 moved from S01 to S02
platform P1 moved from S02 to S01
platform P2 moved from S05 to S02
platform P5 moved from S06 to S05
platform P3 moved from S03 to S06
platform P2 moved from S02 to S03
platform P2 moved from S03 to S02
platform P3 moved from S06 to S03
platform P5 moved from S05 to S06
platform P2 moved from S02 to S05
platform P2 moved from S05 to S02
platform P5 moved from S06 to S05
platform P3 moved from S03 to S06
platform P2 moved from S02 to S03
platform P1 moved from S01 to S02
platform P4 moved from S04 to S01
platform P4 moved from S01 to S04
platform P4 moved from S04 to S01
platform P4 moved from S01 to S04
platform P4 moved from S04 to S01
platform P5 moved from S05 to S04
platform P5 moved from S04 to S05
platform P4 moved from S01 to S04
platform P1 moved from S02 to S01
platform P1 moved from S01 to S02
platform P1 moved from S02 to S01
platform P2 moved from S03 to S02
platform P2 moved from S02 to S03
platform P1 moved from S01 to S02
platform P4 moved from S04 to S01
platform P5 moved from S05 to S04
platform P1 moved from S02 to S05
platform P4 moved from S01 to S02

Car with plate: 103 parked on platform P4

Parking is full

CLIPS>

Επτά αυτοκίνητα – depth

CLIPS> (run)

Parking Begins !

More cars than spaces!

CLIPS>

Β' ΜΕΡΟΣ – Επέκταση προβλήματος

Περιγραφή νέου προβλήματος

Το πρόβλημα του Parking – εκτεταμένη εκδοχή

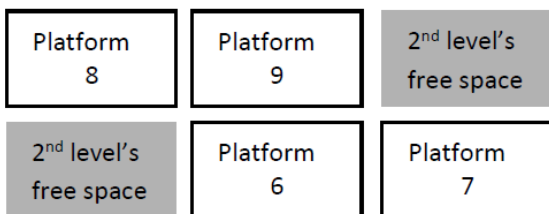
Στην ολοκληρωμένη εκδοχή του προβλήματος, το parking είναι πολυεπίπεδο με 3 επίπεδα (1-3) που το κάθε ένα διαιρείται σε 6 χώρους (spaces) υποδοχής πλατφορμών. Επιπλέον, υπάρχουν 14 πλατφόρμες (platforms), αριθμημένες από P1 έως P14.

Στην αρχή, υπάρχει ένα τουλάχιστον αυτοκίνητο σε αναμονή, δεν υπάρχουν αυτοκίνητα μέσα στο parking και οι πλατφόρμες είναι τοποθετημένες στους υπάρχοντες χώρους όπως αποτυπώνεται στα σχήματα 1-3.

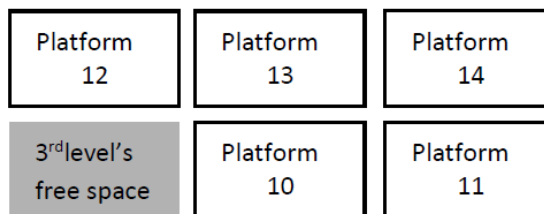
Μια πλατφόρμα μπορεί να κινηθεί σε ένα γειτονικό της χώρο του ίδιου επιπέδου αρκεί αυτός να είναι κενός. Για την δυνατότητα κίνησης μιας πλατφόρμας από όροφο σε όροφο, οι εμπρός κεντρικές θέσεις των ορόφων (πάνω από τον χώρο εισόδου) συγκροτούν ένα φρεάτιο στο οποίο είναι προσαρμοσμένα συρματόσκοινα που διευκολύνουν την κάθετη κίνηση μιας πλατφόρμας (προς τα πάνω ή προς τα κάτω).

Για να μπορέσει ένα αυτοκίνητο να μπει στο parking, θα πρέπει να υπάρχει τουλάχιστον μια ελεύθερη πλατφόρμα σε οποιονδήποτε χώρο οποιουδήποτε επιπέδου του parking και να μετακινηθεί από τον αυτόματο παρκαδόρο στον χώρο εισόδου.

Όταν ένας οδηγός επιθυμεί να παρκάρει το αυτοκίνητό του, δίνει τα στοιχεία του στο αυτόματο παρκαδόρο που φροντίζει να εντοπίσει και να φέρει μια ελεύθερη πλατφόρμα στο χώρο εισόδου. Αντίστοιχα, όταν ένας οδηγός θέλει να παραλάβει το παρκαρισμένο αυτοκίνητό του, δίνει τα στοιχεία του αυτοκινήτου, ο αυτόματος παρκαδόρος εντοπίζει την πλατφόρμα πάνω στην οποία είναι τοποθετημένο το αυτοκίνητο, και αυτή μεταφέρεται αυτόματα και τοποθετείται στο χώρο εισόδου ώστε να παραλάβει ο οδηγός το αυτοκίνητο και να ελευθερώσει την πλατφόρμα.



Σχήμα 2: Κάτοψη 2^{ου} επιπέδου



Σχήμα 3: Κάτοψη 3^{ου} επιπέδου

Υλοποίηση προβλήματος - Απαιτούμενα

Ολοκληρώστε, προσθέτοντας τα γεγονότα που περιγράφουν την Β' έκδοχή του προβλήματος και επιλύστε εκ νέου το ΕΣ. Στη συνέχεια, επιτρέψτε στον χρήστη του συστήματος να εισάγει στην αρχή εκτέλεσης του ΕΣ ένα-ένα τα αυτοκίνητα που θα αναμένουν για είσοδο στο parking δίνοντας τον αριθμό κάθε αυτοκινήτου. Μετά την εισαγωγή των αυτοκινήτων, ο αυτόματος παρκαδόρος θα αρχίσει τη διαδικασία εισαγωγής και τοποθέτησης σε πλατφόρμες.

Μοντελοποίηση

Για την εκτεταμένη εκδοχή του προγράμματος έγιναν οι κατάλληλες αλλαγές, ώστε να προσαρμοστεί στις καινούριες απαιτήσεις του προβλήματος.

Αρχική κατάσταση

Αρχικά, προστίθενται καινούρια δεδομένα - γεγονότα καθώς πλέον υπάρχουν 16 χώροι στάθμευσης και 14 πλατφόρμες που μπορούν να υποδεχτούν κάποιο αυτοκίνητο.

Επιπλέον, το πρόγραμμα φέρει τη δυνατότητα επιλογής από ένα μενού στο χρήστη, ώστε να επιλέξει ποια ενέργεια θέλει ακολουθηθεί. Το μενού αυτό περιλαμβάνει τις επιλογές, αν θέλει να προσθέσει αυτοκίνητο, αν θέλει να αφαιρέσει αυτοκίνητο ή αν θέλει να τερματίσει το πρόγραμμα.

Κατάσταση στόχου

Στο πρόγραμμά μας πλέον, ως στόχους έχουμε να μπουν στο σύστημα του πάρκινγκ τα αυτοκίνητα που βρίσκονται σε κατάσταση αναμονής ή να αφαιρεθούν τα επιθυμούμενα αυτοκίνητα. Τέλος, ως λήξη προγράμματος, θεωρείται η κατάσταση στην οποία ο χρήστης θέλει να αποχωρήσει από το σύστημα πάρκινγκ.

Νέοι Κανόνες

Όπως και στο αρχικό πρόβλημα, το πρόγραμμα ξεκινάει από τον κανόνα `begin` και ύστερα ακολουθεί η εμφάνιση επιλογής μενού. Ανάλογα με την επιθυμία του χρήστη το πρόγραμμα θα ακολουθήσει την κατάλληλη ροή. Επιπλέον, προστέθηκαν κι άλλοι κανόνες ώστε να γίνεται και η μετακίνηση των πλατφορμών μεταξύ των επιπέδων καθώς και κανόνες ελέγχου για την αιτούμενη ενέργεια που επιθυμεί ο χρήστης.

Για το μενού:

- `Choice`
- `Add_car`
- `Want_exit`
- `Exit_parking`
- `Exit_done`

Για τις μετακινήσεις μεταξύ επιπέδων:

- `Move_platform_up_to_1`
- `Move_platform_up_to_2`
- `Move_platform_down_from_2`
- `Move_platform_down_from_1`

Απαραίτητοι έλεγχοι

- `Empty_parking`
- `All_platforms_full`
- `Car_not_exists`

Αναπαράσταση του Νέου Κόσμου Προβλήματος

Τροποποίηση υπάρχων κανόνων

Στη συνέχεια τροποποιήσαμε τους κανόνες από το αρχικό μας πρόβλημα, ώστε να προσαρμοστούν στις καινούριες απαιτήσεις που θέτει η εκτεταμένη εκδοχή.

I. Begin

Έγινε η προσθήκη (`assert (choice add)`) στον κανόνα, ώστε όταν ξεκινήσει το πρόγραμμα, να εισαχθεί και στα γεγονότα ότι επιθυμούμε να μπει ένας αριθμός αυτοκινήτων στο πάρκινγκ.

```
(defrule begin
  (declare (salience 100))
  (initial-fact)
=>
  (set-strategy random)
  (printout t "Parking Begins!" crlf crlf)
  (assert (choice add))
)
```

II. allparked_goal

Ο κανόνας `goal` έχει πλέον τροποποιηθεί και του έχει δοθεί η ονομασία, `allparked_goal`. Ο συγκεκριμένος, όταν ενεργοποιείται γίνεται και η προσθήκη του γεγονότος (`assert (choice menu)`) ώστε η επόμενη ως προς εκτέλεση κανόνα να είναι η επιλογή μενού.

```
(defrule allparked_goal
  (declare (salience 97))
  (not (car (c_plate ?) (c_state waiting)))
  ?x <- (choice add)
=>
  (printout t "All cars waiting have parked" crlf)
  (retract ?x)
  (assert (choice menu))
)
```

Παρουσίαση νέων κανόνων

III. choice

Ο κανόνας `choice` πυροδοτείται όταν στη λίστα των γεγονότων υπάρχει το (`choice menu`). Αν εκτελεσθεί, εκτυπώνει στην κονσόλα ένα μήνυμα προς το χρήστη να επιλέξει

1. Αν επιθυμεί να γίνει εισαγωγή κι άλλων αυτοκινήτων
2. Αν επιθυμεί να αφαιρέσει κάποιο αυτοκίνητο από το σύστημα του πάρκινγκ
3. Αν επιθυμεί να τερματίσει το πρόγραμμα.

```
(defrule choice
(declare (salience 98))
  ?x <- (choice menu)
=>
  (printout t "Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE"
  crlf)
  (bind ?ch(read))
  (if (= ?ch 0) then
    (halt)
  )
  else
    (if (= ?ch 1) then
      (retract ?x)
      (assert (choice add))
    else
      (retract ?x)
      (assert (choice exit))
    )
  )
)
```

Επεξήγηση κώδικα

```
(bind ?ch(read))
```

Διάβασε από το πληκτρολόγιο και αποθήκευσε την τιμή στη μεταβλητή `?ch`.

```
(retract ?x)
(assert (choice add))
```

Αν επιλέξει ο χρήστης την επιλογή 1, τότε θα αφαιρεθεί από τα γεγονότα το `(choice menu)` και θα εισαχθεί το γεγονός `(choice add)` το οποίο στη συνέχεια θα πυροδοτήσει τον κανόνα `enter_parking`.

```
(retract ?x)
(assert (choice exit))
```

Αν επιλέξει ο χρήστης την επιλογή 2, τότε θα αφαιρεθεί από τα γεγονότα το `(choice menu)` και θα εισαχθεί το γεγονός `(choice exit)` το οποίο στη συνέχεια θα πυροδοτήσει τον κανόνα `want_exit`.

IV. `add_car`

Ο κανόνας προσθήκης αυτοκινήτου έχει τη δεύτερη μεγαλύτερη προτεραιότητα με `salience 99`. Ενεργοποιείται όταν βρίσκει στη λίστα γεγονότων, το γεγονός `(choice add)`. Μετά την εκτέλεσή του, ζητά από το χρήστη να εισάγει πόσα αυτοκίνητα θέλουν να μπουν στο πάρκινγκ καθώς και τις πινακίδες τους.

```
(defrule add_car
(declare (salience 99))
  ?x <- (choice add)
=>
```

```
(printout t "Give me the information of the cars waiting" crlf)
(printout t "How many cars are waiting?" crlf)
(bind ?n_cars (read))
(bind ?count 0)
  (while (< ?count ?n_cars) do
    (printout t "Give me the plate of the car:" crlf)
    (assert(car (c_plate(read))))
    (bind ?count (+ ?count 1))
  )
)
```

Παράδειγμα εκτέλεσης κανόνα

```
Give me the information of the cars waiting
How many cars are waiting?
2
Give me the plate of the car:
C01
Give me the plate of the car:
C02
```

V. want_exit

Ο κανόνας εκτελείται όταν ο χρήστης επιλέξει την επιλογή 2, καθώς ταυτοποιείται με το γεγονός (choice exit). Μετά την εκτέλεσή του, ζητά από το χρήστη να εισάγει τον αριθμό αυτοκινήτων που επιθυμούν να βγουν από το πάρκινγκ. Στη συνέχεια, μέσα σε μια επανάληψη, ζητά από το χρήστη να πληκτρολογήσει την πινακίδα του αυτοκινήτου που θέλει να αποχωρήσει από το πάρκινγκ. Η λειτουργία του κανόνα είναι παρόμοια με την enter_parking με τη διαφορά ότι στην κάθε επανάληψη γίνεται η προσθήκη νέων γεγονότων (assert(car (c_plate ?plate) (c_state exiting))).

```
(defrule want_exit
(declare (salience 75))
  ?x <- (choice exit)
=>
  (printout t "How many cars want to exit the parking?" crlf)
  (bind ?n_cars (read))
  (while (> ?n_cars 0)
    (printout t "Give me plate of the car that wants to exit:" crlf)
    (bind ?plate (read))
    (assert(car (c_plate ?plate) (c_state exiting)))
    (bind ?n_cars (- ?n_cars 1))
  )
)
```

Παράδειγμα εκτέλεσης κανόνα

```
How many cars want to exit the parking?  
2  
Give me plate of the car that wants to exit:  
abc003  
Give me plate of the car that wants to exit:  
abc001
```

VI. exit_parking

Ο σκοπός αυτού του κανόνα είναι να εξάγει αυτοκίνητο από το πάρκινγκ. Αν βρει κάποιο αυτοκίνητο που θέλει να βγει στο χώρο υποδοχής, τότε αφαιρεί το αντίστοιχο γεγονός αυτοκινήτου και αλλάζει την κατάσταση της πλατφόρμας σε empty.

```
(defrule exit_parking  
  (declare (salience 96))  
  ?x <- (car (c_plate ?plate) (c_state exiting))  
  ?y <- (car (c_plate ?plate) (c_state ?s))  
  ?z <- (platform (p_name ?s) (p_space S02) (p_state full))  
=>  
  (modify ?z (p_state empty))  
  (retract ?x)  
  (retract ?y)  
  (printout t "Car with plate: " ?plate " exited parking" crlf)  
)
```

VII. exit_done

Με τον κανόνα exit_done επισημαίνεται η επιτυχής λήξη αφαίρεσης των επιθυμητών αυτοκινήτων. Επιπλέον, εισάγει ξανά το γεγονός (choice menu) ώστε να κάνει trigger την επιλογή μενού ξανά και ταυτόχρονα αφαιρεί το γεγονός (choice exit).

```
(defrule exit_done  
  (declare (salience 65))  
  (not (car (c_state exiting)))  
  ?x <- (choice exit)  
=>  
  (printout t "All choosen cars have exited the parking" crlf)  
  (retract ?x)  
  (assert (choice menu))  
)
```

Η επέκταση του προγράμματος απαιτεί και την προσθήκη των επιπλέον επιπέδων πέρα από εκείνο που είχε οριστεί στο αρχικό μας πρόβλημα. Για την υλοποίηση της μετακίνησης μεταξύ των επιπέδων, δημιουργήθηκαν οι παρακάτω κανόνες.

Μεταξύ των ήδη υπάρχοντων κανόνων, προσθέτουμε τις `move_platform_up_to_1`, `move_platform_up_from_2`, `move_platform_down_to_2`, `move_platform_down_to_1`, οι οποίοι έχουν παρόμοια λειτουργία με τη `move_platform_east`.

VIII. `move_platform_up_to_1`

Ο κανόνας, εξασφαλίζει τη μετακίνηση πλατφορμών μεταξύ επιπέδων 0 «s_floor 1» και 1 «s_floor 2». Ψάχνει αν στο χώρο S02 επιπέδου 0, φιλοξενείται κάποια πλατφόρμα και αν στο αντίστοιχο χώρο του επιπέδου 1 δεν περιέχει πλατφόρμα. Επίσης, κρατά και την πληροφορία κατάστασης της πλατφόρμας που υπάρχει στον S02 επιπέδου 0. Αν εκτελεστεί ο κανόνας, τότε γίνονται οι κατάλληλες τροποποιήσεις στις καταστάσεις των χώρων καθώς και σε ποιο επίπεδο βρίσκεται πλέον η πλατφόρμα.

```
(defrule move_platform_up_to_1
(declare (salience 50))
  ?x <- (space (s_name S02) (s_floor 1) (s_state P))
  ?p <- (platform (p_name ?pname) (p_space S02) (p_state ?))
  ?y <- (space (s_name S12) (s_floor 2) (s_state F))
=>
  (modify ?x (s_state F))
  (modify ?p (p_space S12))
  (modify ?y (s_state P))
  (printout t "platform " ?pname " moved from S02 to S12" crlf)
)
```

Παράδειγμα εκτέλεσης κανόνα

Ως προς πυροδότηση κανόνα με τα παρακάτω γεγονότα

```
?x <- (space (s_name S02) (s_floor 1) (s_state P))
?p <- (platform (p_name P3) (p_space S02) (p_state parked))
?y <- (space (s_name S12) (s_floor 2) (s_state F))
```

Μετά την εκτέλεση του κανόνα

```
(space (s_name S02) (s_floor 1) (s_state F))
(platform (p_name P3) (p_space S12) (p_state parked))
(space (s_name S12) (s_floor 2) (s_state P))
```


IX. move_platform_up_to_2

Αντίστοιχα, οι παρακάτω 3 κανόνες λειτουργούν με τον παρόμοιο τρόπο ανάλογα με τα επίπεδα που αναφέρονται.

```
(defrule move_platform_up_to_2
(declare (salience 50))
  ?x <- (space (s_name S12) (s_floor 2) (s_state P))
  ?p <- (platform (p_name ?pname) (p_space S12) (p_state ?))
  ?y <- (space (s_name S22) (s_floor 3) (s_state F))
=>
  (modify ?x (s_state F))
  (modify ?p (p_space S22))
  (modify ?y (s_state P))
  (printout t "platform " ?pname " moved from S12 to S22" crlf)
)
```

X. move_platform_down_from_2

Μετακίνηση, μεταξύ επιπέδου 2 «s_floor 3» και επιπέδου 1 «s_floor 2». Παρατηρείστε ότι και στις 4 περιπτώσεις, έχουμε βάλει την ίδια προτεραιότητα με (salience 50).

```
(defrule move_platform_down_from_2
(declare (salience 50))
  ?x <- (space (s_name S22) (s_floor 3) (s_state P))
  ?p <- (platform (p_name ?pname) (p_space S22) (p_state ?))
  ?y <- (space (s_name S12) (s_floor 2) (s_state F))
=>
  (modify ?x (s_state F))
  (modify ?p (p_space S12))
  (modify ?y (s_state P))
  (printout t "platform " ?pname " moved from S22 to S12" crlf)
)
```

XI. move_platform_down_from_1

```
(defrule move_platform_down_from_1
(declare (salience 50))
  ?x <- (space (s_name S12) (s_floor 2) (s_state P))
  ?p <- (platform (p_name ?pname) (p_space S12) (p_state ?))
  ?y <- (space (s_name S02) (s_floor 1) (s_state F))
=>
  (modify ?x (s_state F))
  (modify ?p (p_space S02))
  (modify ?y (s_state P))
  (printout t "platform " ?pname " moved from S12 to S02" crlf)
)
```

XII. empty_parking

Οι κανόνες που ακολουθούνται είναι κανόνες ελέγχου τους οποίους θεωρήσαμε εμείς απαραίτητους για τη βελτιστοποίηση εκτέλεσης του προγράμματος.

Ο κανόνας `empty_parking`, ελέγχει την περίπτωση που ο χρήστης επιλέξει να αφαιρεθεί αυτοκίνητο από το πάρκινγκ ενώ αυτός είναι άδειος. Αν βρεθεί σε τέτοια κατάσταση, τότε τον σταματά και τον επιστρέφει ξανά στην κατάσταση επιλογής μενού.

```
(defrule empty_parking
(declare (salience 70))
  (not (exists (car)))
  ?x <- (choice exit)
=>
  (printout t "The parking is empty!" crlf)
  (retract ?x)
  (assert (choice menu))
)
```

XIII. all_platforms_full

Αντίστοιχα, υπάρχει ο κανόνας `all_platforms_full`. Ο σκοπός του είναι να εξετάσει την περίπτωση εισαγωγής επιπλέον αυτοκινήτων, ενώ το σύστημα πάρκινγκ είναι γεμάτο.

```
(defrule all_platforms_full
(declare (salience 70))
  (not (platform (p_state empty)))
  ?x <- (choice add)
=>
  (printout t "All platforms are full!" crlf)
  (retract ?x)
  (assert (choice menu))
)
```

XIV. car_not_exists

Τέλος, για την ορθότερη λειτουργία του προγράμματος ορίζεται και ο κανόνας `car_not_exists`. Ο συγκεκριμένος κανόνας ελέγχει αν το αυτοκίνητο που επιθυμεί ο χρήστης να αφαιρεθεί δεν υπάρχει στο σύστημα του πάρκινγκ.

```
(defrule car_not_exists
(declare (salience 80))
  ?x <- (car (c_plate ?plate) (c_state exiting))
  (not(car (c_plate ?plate) (c_state ?s&~exiting)))
  (choice exit)
=>
  (printout t "The car with plate " ?plate " is not in the parking" crlf)
  (retract ?x)
)
```

Πρότυπος κώδικας

```
// -----  
//                               ΒΕΛΑΣΚΟ ΠΑΟΛΑ           cs161020  
//                               ΜΙΧΑ ΕΥΑΓΓΕΛΙΑ           cs171102  
// -----  
//                               Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών  
//                               Εργαστήριο TN 2020/21 - Εργασία 2: Εμπειρα Συστήματα  
//                               Το πρόβλημα του πάρκινγκ - Ολοκληρωμένη εκδοχή  
// -----  
// Υπεύθυνος μαθήματος ΓΕΩΡΓΟΥΛΗ ΚΑΤΕΡΙΝΑ  
// Καθηγητές: ΒΟΥΛΟΔΗΜΟΣ ΘΑΝΟΣ, ΤΣΕΛΕΝΤΗ ΠΑΝΑΓΙΩΤΑ  
// -----  
  
// -----  
//                               Πρότυπα Γεγονότων  
// -----  
  
(deftemplate space  
  (slot s_name (type SYMBOL))  
  (slot s_floor (type INTEGER))  
  (slot s_state (type SYMBOL))  
)  
  
(deftemplate platform  
  (slot p_name (type SYMBOL))  
  (slot p_space (type SYMBOL))  
  (slot p_state (type SYMBOL))  
)  
  
(deftemplate car  
  (slot c_plate (type SYMBOL))  
  (slot c_state (type SYMBOL) (default waiting))  
)  
  
// -----  
//                               Αρχικά Γεγονότα  
// -----  
  
(def facts spaces "Γεγονότα: χώροι στάθμευσης"  
  (space (s_name S01) (s_floor 1) (s_state P))  
  (space (s_name S02) (s_floor 1) (s_state F))  
  (space (s_name S03) (s_floor 1) (s_state P))  
  (space (s_name S04) (s_floor 1) (s_state P))  
  (space (s_name S05) (s_floor 1) (s_state P))  
  (space (s_name S06) (s_floor 1) (s_state P))  
  (space (s_name S11) (s_floor 2) (s_state F))  
  (space (s_name S12) (s_floor 2) (s_state P))  
  (space (s_name S13) (s_floor 2) (s_state P))  
  (space (s_name S14) (s_floor 2) (s_state P))  
  (space (s_name S15) (s_floor 2) (s_state P))  
  (space (s_name S16) (s_floor 2) (s_state F))
```

```
(space (s_name S21) (s_floor 3) (s_state F))
(space (s_name S22) (s_floor 3) (s_state P))
(space (s_name S23) (s_floor 3) (s_state P))
(space (s_name S24) (s_floor 3) (s_state P))
(space (s_name S25) (s_floor 3) (s_state P))
(space (s_name S26) (s_floor 3) (s_state P))
)

(deffacts platforms "Γεγονότα: πλατφόρμες"
  (platform (p_name P1) (p_space S01) (p_state empty))
  (platform (p_name P2) (p_space S03) (p_state empty))
  (platform (p_name P3) (p_space S04) (p_state empty))
  (platform (p_name P4) (p_space S05) (p_state empty))
  (platform (p_name P5) (p_space S06) (p_state empty))
  (platform (p_name P6) (p_space S12) (p_state empty))
  (platform (p_name P7) (p_space S13) (p_state empty))
  (platform (p_name P8) (p_space S14) (p_state empty))
  (platform (p_name P9) (p_space S15) (p_state empty))
  (platform (p_name P10) (p_space S22) (p_state empty))
  (platform (p_name P11) (p_space S23) (p_state empty))
  (platform (p_name P12) (p_space S24) (p_state empty))
  (platform (p_name P13) (p_space S25) (p_state empty))
  (platform (p_name P14) (p_space S26) (p_state empty))
)

(deffacts neighbours "Γεγονότα: γειτνίαση των χώρων "
  ;;; Επίπεδο 0
  (east S01 S02)
  (east S02 S03)
  (east S04 S05)
  (east S05 S06)
  (north S01 S04)
  (north S02 S05)
  (north S03 S06)

  ;;; Επίπεδο 1
  (east S11 S12)
  (east S12 S13)
  (east S14 S15)
  (east S15 S16)
  (north S11 S14)
  (north S12 S15)
  (north S13 S16)

  ;;; Επίπεδο 2
  (east S21 S22)
  (east S22 S23)
  (east S24 S25)
  (east S25 S26)
  (north S21 S24)
  (north S22 S25)
  (north S23 S26)
)
```

```
;Γεγονός: πλήθος αυτοκινήτων που μπορεί να υποδεχτεί το πάρκινγκ
(deffacts initial
  (capacity 14)
)

;; -----
;;                               Κανόνες
;; -----

(defrule begin "Αρχικοποίηση στρατηγικής και έναρξης προγράμματος"
(declare (salience 100))
  (initial-fact)
=>
;;;;;----- Ανάθεση στρατηγικής "random" -----
  (set-strategy random)
  ;; (set-strategy depth)
  ;; (set-strategy breadth)
  ;; (set-strategy simplicity)
  ;; (set-strategy complexity)
  (printout t "Parking Begins!" crlf crlf)
  (assert (choice add))
)

(defrule choice "Μενού επιλογής"
(declare (salience 98))
  ?x <- (choice menu)
=>
  (printout t "Type 1. ADD car(s), 2. REMOVE car(s) or 0. TERMINATE" crlf)
  (bind ?ch(read))
  (if (= ?ch 0) then
    (halt)
  )
  else
    (if (= ?ch 1) then
      (retract ?x)
      (assert (choice add))
    )
    else
      (retract ?x)
      (assert (choice exit))
    )
  )
)

(defrule check_capacity "Έλεγχος εγκυρότητας εισαγωγή αριθμό αυτοκινήτων"
(declare (salience 99))
  ?x <- (car (c_plate ?) (c_state ?))
  (capacity ?cap)
  (test (> (length (find-all-facts ((?c car)) (eq ?c:c_state waiting) ) )
?cap ) )
=>
  (printout t "More cars than spaces!" crlf)
  (halt)
)
```

```
(defrule add_car "Προσθήκη αυτοκινήτων"
(declare (salience 99))
  ?x <- (choice add)
=>
  (printout t "Give me the information of the cars waiting" crlf)
  (printout t "How many cars are waiting?" crlf)
  (bind ?n_cars (read))
  (bind ?count 0)
  (while (< ?count ?n_cars) do
    (printout t "Give me the plate of the car:" crlf)
    (assert(car (c_plate(read))))
    (bind ?count (+ ?count 1))
  )
)

(defrule enter_parking "Είσοδος αυτοκινήτου"
(declare (salience 98))
  ?x <- (car (c_plate ?plate) (c_state waiting))
  ?y <- (platform (p_name ?pname) (p_space S02) (p_state empty))
=>
  (modify ?x (c_state ?pname))
  (modify ?y (p_state full))
  (printout t "Car with plate: " ?plate " parked on platform " ?pname crlf)
)

(defrule allparked_goal "Εύρεση στόχου - Εισαγωγή όλων των επιθυμητών αυτοκινήτων"
(declare (salience 97))
  (not (car (c_plate ?) (c_state waiting)))
  ?x <- (choice add)
=>
  (printout t "All cars waiting have parked" crlf)
  (retract ?x)
  (assert (choice menu))
)

(defrule want_exit "Επιλογή εξόδου αυτοκινήτου"
(declare (salience 75))
  ?x <- (choice exit)
=>
  (printout t "How many cars want to exit the parking?" crlf)
  (bind ?n_cars (read))
  (while (> ?n_cars 0)
    (printout t "Give me plate of the car that wants to exit:" crlf)
    (bind ?plate (read))
    (assert(car (c_plate ?plate) (c_state exiting)))
    (bind ?n_cars (- ?n_cars 1))
  )
)

(defrule exit_parking "Αυτοκίνητα προς έξοδο"
```

```
(declare (salience 96))
?x <- (car (c_plate ?plate) (c_state exiting))
?y <- (car (c_plate ?plate) (c_state ?s))
?z <- (platform (p_name ?s) (p_space S02) (p_state full))
=>
(modify ?z (p_state empty))
(retract ?x)
(retract ?y)
(printout t "Car with plate: " ?plate " exited parking" crlf)
)

(defrule exit_done "Εύρεση στόχου - Εξαγωγή όλων των επιθυμητών αυτοκινήτων"
  (declare (salience 65))
  (not (car (c_state exiting)))
  ?x <- (choice exit)
=>
  (printout t "All choosen cars have exited the parking" crlf)
  (retract ?x)
  (assert (choice menu))
)

(defrule empty_parking "Έλεγχος διαθεσιμότητας χώρου πάρκινγκ - ΑΔΕΙΟΣ"
  (declare (salience 70))
  (not (exists (car)))
  ?x <- (choice exit)
=>
  (printout t "The parking is empty!" crlf)
  (retract ?x)
  (assert (choice menu))
)

(defrule all_platforms_full "Έλεγχος διαθεσιμότητας χώρου πάρκινγκ - ΓΕΜΑΤΟΣ"
  (declare (salience 70))
  (not (platform (p_state empty)))
  ?x <- (choice add)
=>
  (printout t "All platforms are full!" crlf)
  (retract ?x)
  (assert (choice menu))
)

(defrule car_not_exists "Έλεγχος αυτοκινήτου"
  (declare (salience 80))
  ?x <- (car (c_plate ?plate) (c_state exiting))
  (not (car (c_plate ?plate) (c_state ?s&~exiting)))
  (choice exit)
=>
  (printout t "The car with plate " ?plate " is not in the parking" crlf)
  (retract ?x)
)

(defrule move_platform_east "Μετακίνηση πλατφόρμας δεξιά"
```

```
(declare (salience 50))
  (east ?s ?s_right)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_right) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_right))
  (printout t "platform " ?pname " moved from " ?s " to " ?s_right crlf)
)

(defrule move_platform_west "Μετακίνηση πλατφόρμας αριστερά"
(declare (salience 50))
(east ?s_left ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_left) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_left))
  (printout t "platform " ?pname " moved from " ?s " to " ?s_left crlf)
)

(defrule move_platform_north "Μετακίνηση πλατφόρμας πάνω"
(declare (salience 50))
  (north ?s ?s_above)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_above) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_above))
  (printout t "platform " ?pname " moved from " ?s " to " ?s_above crlf)
)

(defrule move_platform_south "Μετακίνηση πλατφόρμας κάτω"
(declare (salience 50))
  (north ?s_down ?s)
  ?x <- (space (s_name ?s) (s_floor ?) (s_state P))
  ?y <- (space (s_name ?s_down) (s_floor ?) (s_state F))
  ?z <- (platform (p_name ?pname) (p_space ?s) (p_state ?))
=>
  (modify ?x (s_state F))
  (modify ?y (s_state P))
  (modify ?z (p_space ?s_down))
  (printout t "platform " ?pname " moved from " ?s " to " ?s_down crlf)
)

(defrule move_platform_up_to_1 "Μετακίνηση πλατφόρμας από επίπεδο 0 στο
επίπεδο 1"
```



```
(declare (salience 50))
?x <- (space (s_name S02) (s_floor 1) (s_state P))
?p <- (platform (p_name ?pname) (p_space S02) (p_state ?))
?y <- (space (s_name S12) (s_floor 2) (s_state F))
=>
(modify ?x (s_state F))
(modify ?p (p_space S12))
(modify ?y (s_state P))
(printout t "platform " ?pname " moved from S02 to S12" crlf)
)

(defrule move_platform_up_to_2 "Μετακίνηση πλατφόρμας από επίπεδο 1 στο
επίπεδο 2"
(declare (salience 50))
?x <- (space (s_name S12) (s_floor 2) (s_state P))
?p <- (platform (p_name ?pname) (p_space S12) (p_state ?))
?y <- (space (s_name S22) (s_floor 3) (s_state F))
=>
(modify ?x (s_state F))
(modify ?p (p_space S22))
(modify ?y (s_state P))
(printout t "platform " ?pname " moved from S12 to S22" crlf)
)

(defrule move_platform_down_from_2 "Μετακίνηση πλατφόρμας από επίπεδο 2 στο
επίπεδο 1"
(declare (salience 50))
?x <- (space (s_name S22) (s_floor 3) (s_state P))
?p <- (platform (p_name ?pname) (p_space S22) (p_state ?))
?y <- (space (s_name S12) (s_floor 2) (s_state F))
=>
(modify ?x (s_state F))
(modify ?p (p_space S12))
(modify ?y (s_state P))
(printout t "platform " ?pname " moved from S22 to S12" crlf)
)

(defrule move_platform_down_from_1 "Μετακίνηση πλατφόρμας από επίπεδο 1 στο
επίπεδο 0"
(declare (salience 50))
?x <- (space (s_name S12) (s_floor 2) (s_state P))
?p <- (platform (p_name ?pname) (p_space S12) (p_state ?))
?y <- (space (s_name S02) (s_floor 1) (s_state F))
=>
(modify ?x (s_state F))
(modify ?p (p_space S02))
(modify ?y (s_state P))
(printout t "platform " ?pname " moved from S12 to S02" crlf)
)
```

Αποτελέσματα Εκτέλεσης

Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική simplicity

Ατέρμονος

Επιτυχής εισαγωγή 2 αυτοκινήτων μέχρι να βρεθεί σε κατάσταση ατέρμονου βρόχου.

Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική complexity

Ατέρμονος

Επιτυχής εισαγωγή 3 αυτοκινήτων μέχρι να βρεθεί σε κατάσταση ατέρμονου βρόχου.

Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική depth

Ατέρμονος

Επιτυχής εισαγωγή ενός αυτοκινήτου

Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική breadth

Ατέρμονος

Δεν εισάγεται ούτε ένα αυτοκίνητο.

Εισαγωγή 4 αυτοκινήτων – Εξαγωγή 2 αυτοκινήτων: στρατηγική random

CLIPS> (run)

Parking Begins!

Give me the information of the cars waiting

How many cars are waiting?

4

Give me the plate of the car:

abc001

Give me the plate of the car:

abc002

Give me the plate of the car:

abc003

Give me the plate of the car:

abc004

platform P4 moved from S05 to S02

Car with plate: abc004 parked on platform P4

platform P12 moved from S24 to S21

platform P7 moved from S13 to S16

platform P13 moved from S25 to S24

platform P13 moved from S24 to S25

platform P13 moved from S25 to S24

platform P10 moved from S22 to S25

platform P6 moved from S12 to S22

platform P6 moved from S22 to S12

platform P10 moved from S25 to S22

platform P10 moved from S22 to S25

platform P10 moved from S25 to S22

platform P14 moved from S26 to S25

platform P11 moved from S23 to S26

platform P10 moved from S22 to S23

platform P10 moved from S23 to S22

platform P6 moved from S12 to S11

platform P4 moved from S02 to S12

platform P4 moved from S12 to S13

platform P9 moved from S15 to S12

platform P8 moved from S14 to S15

platform P6 moved from S11 to S14

platform P9 moved from S12 to S02

Car with plate: abc001 parked on platform P9

platform P9 moved from S02 to S05

platform P10 moved from S22 to S12

platform P14 moved from S25 to S22

platform P11 moved from S26 to S25

platform P10 moved from S12 to S11

platform P2 moved from S03 to S02

Car with plate: abc002 parked on platform P2

platform P5 moved from S06 to S03

platform P2 moved from S02 to S12

platform P9 moved from S05 to S02

platform P9 moved from S02 to S05

platform P9 moved from S05 to S06

```
platform P5 moved from S03 to S02
Car with plate: abc003 parked on platform P5
All cars waiting have parked
Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE
2
How many cars want to exit the parking?
2
Give me plate of the car that wants to exit:
abc003
Give me plate of the car that wants to exit:
abc001
Car with plate: abc003 exited parking
platform P5 moved from S02 to S03
platform P5 moved from S03 to S02
platform P5 moved from S02 to S05
platform P2 moved from S12 to S02
platform P8 moved from S15 to S12
platform P6 moved from S14 to S15
platform P6 moved from S15 to S14
platform P7 moved from S16 to S15
platform P4 moved from S13 to S16
platform P4 moved from S16 to S13
platform P4 moved from S13 to S16
platform P4 moved from S16 to S13
platform P4 moved from S13 to S16
platform P8 moved from S12 to S13
platform P7 moved from S15 to S12
platform P4 moved from S16 to S15
platform P14 moved from S22 to S23
platform P11 moved from S25 to S22
platform P14 moved from S23 to S26
platform P14 moved from S26 to S23
platform P14 moved from S23 to S26
platform P14 moved from S26 to S25
platform P11 moved from S22 to S23
platform P14 moved from S25 to S22
platform P13 moved from S24 to S25
platform P13 moved from S25 to S26
```

platform P12 moved from S21 to S24
platform P14 moved from S22 to S21
platform P7 moved from S12 to S22
platform P4 moved from S15 to S12
platform P12 moved from S24 to S25
platform P4 moved from S12 to S15
platform P8 moved from S13 to S12
platform P12 moved from S25 to S24
platform P13 moved from S26 to S25
platform P13 moved from S25 to S26
platform P12 moved from S24 to S25
platform P12 moved from S25 to S24
platform P13 moved from S26 to S25
platform P13 moved from S25 to S26
platform P13 moved from S26 to S25
platform P11 moved from S23 to S26
platform P7 moved from S22 to S23
platform P14 moved from S21 to S22
platform P12 moved from S24 to S21
platform P13 moved from S25 to S24
platform P14 moved from S22 to S25
platform P8 moved from S12 to S22
platform P2 moved from S02 to S12
platform P5 moved from S05 to S02
platform P2 moved from S12 to S13
platform P5 moved from S02 to S12
platform P9 moved from S06 to S05
platform P9 moved from S05 to S06
platform P9 moved from S06 to S03
platform P9 moved from S03 to S06
platform P9 moved from S06 to S05
platform P9 moved from S05 to S06
platform P5 moved from S12 to S02
platform P2 moved from S13 to S12
platform P2 moved from S12 to S13
platform P5 moved from S02 to S12
platform P2 moved from S13 to S16
platform P2 moved from S16 to S13

platform P4 moved from S15 to S16
platform P5 moved from S12 to S15
platform P5 moved from S15 to S12
platform P5 moved from S12 to S15
platform P8 moved from S22 to S12
platform P8 moved from S12 to S02
platform P10 moved from S11 to S12
platform P10 moved from S12 to S11
platform P8 moved from S02 to S12
platform P14 moved from S25 to S22
platform P14 moved from S22 to S25
platform P12 moved from S21 to S22
platform P12 moved from S22 to S21
platform P12 moved from S21 to S22
platform P1 moved from S01 to S02
platform P1 moved from S02 to S03
platform P8 moved from S12 to S02
platform P2 moved from S13 to S12
platform P13 moved from S24 to S21
platform P13 moved from S21 to S24
platform P12 moved from S22 to S21
platform P14 moved from S25 to S22
platform P14 moved from S22 to S25
platform P14 moved from S25 to S22
platform P11 moved from S26 to S25
platform P7 moved from S23 to S26
platform P8 moved from S02 to S05
platform P8 moved from S05 to S02
platform P9 moved from S06 to S05
platform P9 moved from S05 to S06
platform P8 moved from S02 to S05
platform P1 moved from S03 to S02
platform P9 moved from S06 to S03
platform P1 moved from S02 to S01
platform P9 moved from S03 to S02
Car with plate: abc001 exited parking
All choosen cars have exited the parking
Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE

0

CLIPS>

Εισαγωγή 5 αυτοκινήτων – Εξαγωγή 3 αυτοκινήτων: στρατηγική random

(Ολοκληρωμένη εκτέλεση στο αρχείο – random_Insert_05_Exit_03.txt)

CLIPS> Loading Selection...

CLIPS> (reset)

CLIPS> (run)

Parking Begins!

Give me the information of the cars waiting

How many cars are waiting?

5

Give me the plate of the car:

C01

Give me the plate of the car:

C02

Give me the plate of the car:

C03

Give me the plate of the car:

C04

Give me the plate of the car:

C05

platform P9 moved from S15 to S16

platform P8 moved from S14 to S15

platform P8 moved from S15 to S14

platform P6 moved from S12 to S15

platform P6 moved from S15 to S12

platform P6 moved from S12 to S15

platform P8 moved from S14 to S11

platform P8 moved from S11 to S12

platform P8 moved from S12 to S02

Car with plate: C05 parked on platform P8

platform P6 moved from S15 to S14

platform P6 moved from S14 to S11

platform P6 moved from S11 to S14

platform P6 moved from S14 to S15

platform P6 moved from S15 to S12
platform P9 moved from S16 to S15
platform P9 moved from S15 to S14
platform P9 moved from S14 to S11
platform P9 moved from S11 to S14
platform P9 moved from S14 to S15
platform P9 moved from S15 to S16
platform P9 moved from S16 to S15
platform P7 moved from S13 to S16
platform P6 moved from S12 to S13
platform P8 moved from S02 to S12
platform P4 moved from S05 to S02
Car with plate: C04 parked on platform P4
platform P4 moved from S02 to S05
platform P2 moved from S03 to S02
Car with plate: C01 parked on platform P2
platform P2 moved from S02 to S03
platform P8 moved from S12 to S02
platform P6 moved from S13 to S12
platform P7 moved from S16 to S13
platform P9 moved from S15 to S14
platform P9 moved from S14 to S11
platform P9 moved from S11 to S14
platform P6 moved from S12 to S11
platform P8 moved from S02 to S12
platform P1 moved from S01 to S02
Car with plate: C02 parked on platform P1
platform P8 moved from S12 to S15
platform P1 moved from S02 to S12
platform P2 moved from S03 to S02
platform P2 moved from S02 to S03
platform P4 moved from S05 to S02
platform P4 moved from S02 to S01
platform P4 moved from S01 to S02
platform P3 moved from S04 to S01
platform P3 moved from S01 to S04
platform P4 moved from S02 to S01
platform P1 moved from S12 to S02

platform P7 moved from S13 to S12
platform P7 moved from S12 to S13
platform P7 moved from S13 to S12
platform P7 moved from S12 to S13
platform P10 moved from S22 to S12
platform P3 moved from S04 to S05
platform P3 moved from S05 to S04
platform P3 moved from S04 to S05
platform P3 moved from S05 to S04
platform P11 moved from S23 to S22
platform P11 moved from S22 to S23
platform P1 moved from S02 to S05
platform P2 moved from S03 to S02
platform P2 moved from S02 to S03
platform P2 moved from S03 to S02
platform P5 moved from S06 to S03
platform P5 moved from S03 to S06
platform P5 moved from S06 to S03
platform P1 moved from S05 to S06
platform P3 moved from S04 to S05
platform P4 moved from S01 to S04
platform P2 moved from S02 to S01
platform P2 moved from S01 to S02
platform P2 moved from S02 to S01
platform P3 moved from S05 to S02
Car with plate: C03 parked on platform P3
All cars waiting have parked
Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE
2
How many cars want to exit the parking?
2
Give me plate of the car that wants to exit:
C02
Give me plate of the car that wants to exit:
C03
Car with plate: C03 exited parking
platform P11 moved from S23 to S22
platform P14 moved from S26 to S23

platform P14 moved from S23 to S26

platform P11 moved from S22 to S23

-
-
-

platform P13 moved from S16 to S15

platform P9 moved from S12 to S11

platform P13 moved from S15 to S12

platform P13 moved from S12 to S15

platform P4 moved from S02 to S12

platform P1 moved from S05 to S02

Car with plate: C02 exited parking

All choosen cars have exited the parking

Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE

2

How many cars want to exit the parking?

1

Give me plate of the car that wants to exit:

C01

platform P9 moved from S11 to S14

platform P4 moved from S12 to S11

platform P4 moved from S11 to S12

platform P9 moved from S14 to S11

platform P13 moved from S15 to S14

platform P13 moved from S14 to S15

-
-
-

platform P1 moved from S05 to S06

platform P9 moved from S12 to S02

platform P9 moved from S02 to S12

platform P2 moved from S03 to S02

Car with plate: C01 exited parking

All choosen cars have exited the parking

Type 1 to ADD car, 2 for car to EXIT or 0 to TERMINATE

0

CLIPS>

Εισαγωγή 17 αυτοκινήτων: στρατηγική random

CLIPS> (reset)

CLIPS> (run)

Parking Begins!

Give me the information of the cars waiting

How many cars are waiting?

17

Give me the plate of the car:

01

Give me the plate of the car:

02

Give me the plate of the car:

03

Give me the plate of the car:

04

Give me the plate of the car:

05

Give me the plate of the car:

06

Give me the plate of the car:

07

Give me the plate of the car:

08

Give me the plate of the car:

09

Give me the plate of the car:

10

Give me the plate of the car:

11

Give me the plate of the car:

12

Give me the plate of the car:

13

Give me the plate of the car:

14

Give me the plate of the car:

15

Give me the plate of the car:

16

Give me the plate of the car:

17

More cars than spaces!

CLIPS>

Συμπεράσματα – Παρατηρήσεις

Παρατηρήσεις

Έχοντας πλέον διεκπεραιώσει και τη δεύτερη εργαστηριακή άσκηση, μπορούμε να αντιληφθούμε καλύτερα τη λειτουργία ενός έμπειρου συστήματος. Στην εργασία που μας δόθηκε, κατανοήσαμε και τη σημασία της διαφοράς μεταξύ των εργαλείων της CLIPS με τα γνωστά συμβατικά ακολουθιακά-αντικειμενοστρεφή προγράμματα.

Όπως μας είναι ήδη γνωστό η CLIPS προσφέρει επτά διαφορετικούς τρόπους στρατηγικής depth, breadth, LEX, MEA, complexity, simplicity, and random, το καθένα με το δικά του πλεονεκτήματα και μειονεκτήματα. Σε κάθε χρονική στιγμή μόνο μια από αυτές είναι ενεργή και χρησιμοποιείται για την επιλογή του κανόνα από την ατζέντα.

Στη στρατηγική depth οι νέοι κανόνες μπαίνουν "πάνω" από τους "παλαιούς". Η σύγκριση γίνεται με βάση το πότε εισήχθηκαν τα γεγονότα που ικανοποιούν τον κανόνα στη λίστα γεγονότων. Αντίστοιχα στη στρατηγική breadth οι νέοι κανόνες μπαίνουν "κάτω" από τους "παλαιούς".

Στη στρατηγική simplicity οι κανόνες με απλούστερες συνθήκες κατατάσσονται "πάνω" από τους κανόνες με τις πιο πολύπλοκες. Η πολυπλοκότητα ενός κανόνα εξαρτάται από τον αριθμό των συνθηκών και από τις συγκρίσεις (περιορισμούς) που λαμβάνουν χώρα στις συνθήκες. Αντίστοιχα με τη στρατηγική complexity οι πιο πολύπλοκοι κανόνες μπαίνουν "πάνω" από τους "απλούστερους".

Στη στρατηγική LEX οι κανόνες οι οποίοι ενεργοποιούνται από "νεότερα" γεγονότα κατατάσσονται υψηλότερα στη agenda. Για τους κανόνες οι οποίοι κατατάσσονται στην "ίδια ομάδα" με βάση το προηγούμενο κριτήριο, υψηλότερα κατατάσσονται οι κανόνες οι οποίοι έχουν περισσότερες συνθήκες, δηλαδή αποτελεί συνδυασμό των στρατηγικών depth και complexity.

Στη στρατηγική MEA εξετάζεται το γεγονός το οποίο αντιστοιχεί στην πρώτη συνθήκη και οι κανόνες διατάσσονται με βάση το πότε εισήχθηκε αυτό στη λίστα. Όσο νεότερο είναι το γεγονός τόσο "ψηλότερα" μπαίνει ο κανόνας. Για κανόνες οι οποίοι έχουν την ίδια σειρά με βάση αυτό το κριτήριο χρησιμοποιείται η στρατηγική LEX.

Τέλος στη στρατηγική random οι κανόνες μπαίνουν στην ατζέντα με τυχαίο τρόπο. Μετά από διαφορετικά τρεξίματα του προγράμματος που υλοποιήσαμε, παρατηρήσαμε ότι η βέλτιστη στρατηγική σε συνδυασμό με τις τιμές των salience των κανόνων είναι η random. Ενώ, πολλές φορές, ανάλογα με τα δεδομένα που του εισάγουμε, οι υπόλοιπες στρατηγικές είτε έκαναν πολλές περισσότερες άσκοπες κινήσεις ή και ακόμη οδηγούταν σε κατάσταση ατέρμονου βρόχου.

Συμπεράσματα

Ένα Έμπειρο Σύστημα (Expert System) είναι ένα σύστημα σχεδιασμένο για να λύνει δύσκολα προβλήματα που αφορούν την λήψη αποφάσεων και βασίζονται σε γνώση που έχει συγκεντρωθεί από ειδήμονες. Ασχολούνται με προβλήματα ρεαλιστικής περιπλοκότητας, για τα οποία υπάρχει εκτενής ανθρώπινη εμπειρία και γνώση. Οφείλουν να είναι ικανά για αξιολόγηση, επεξήγηση και δικαιολόγηση των λύσεων και των συμβουλών που παρέχουν και τέλος καλούνται να είναι γρήγορα και αξιόπιστα ώστε να είναι χρήσιμα ως εργαλεία.

Οι διαφορές των Έμπειρων Συστημάτων με τα συμβατικά προγράμματα είναι πολλές και καθοριστικές. Αρχικά, τα ΕΣ προσομοιώνουν τον τρόπο επίλυσης ενός προβλήματος ενώ ένα πρόγραμμα προσομοιώνει το ίδιο το πρόβλημα. Τα ΕΣ κάνουν χρήση ευριστικών μεθόδων για τον περιορισμό του χώρου αναζήτησης σε αντίθεση με τα συμβατικά προγράμματα που κάνουν χρήση αλγορίθμων. Επιπλέον, οι γλώσσες που χρησιμοποιούνται στα ΕΣ είναι πιο κοντά στην ανθρώπινη γλώσσα και τρόπο σκέψης ενώ τα προγράμματα χρησιμοποιούν γλώσσες πιο κοντά στο τρόπο λειτουργίας του υπολογιστή. Σε περίπτωση που χρειαστεί αναθεώρηση της γνώσης ενός προβλήματος τότε χρειάζονται μεγάλες αλλαγές σε ένα συμβατικό πρόγραμμα ενώ σε ένα ΕΣ είναι πιο εύκολη και η αλλαγή της γνώσης καθώς και η επέκταση του ίδιου του προβλήματος. Τέλος, πολύ σημαντικό χαρακτηριστικό των Έμπειρων Συστημάτων που στερούνται τα συμβατικά προγράμματα είναι η επεξήγηση του συλλογισμού που οδήγησε στη λύση του προβλήματος.

Παράρτημα

Στρατηγική random

Παρακάτω βλέπουμε ενδεικτική εκτέλεση με τρία αυτοκίνητα και random στρατηγική.

```
CLIPS> (reset)
<== f-0      (initial-fact)
==> f-0      (initial-fact)
==> f-1      (space (s_name S01) (s_floor 1) (s_state P))
==> f-2      (space (s_name S02) (s_floor 1) (s_state F))
==> f-3      (space (s_name S03) (s_floor 1) (s_state P))
==> f-4      (space (s_name S04) (s_floor 1) (s_state P))
==> f-5      (space (s_name S05) (s_floor 1) (s_state P))
==> f-6      (space (s_name S06) (s_floor 1) (s_state P))
==> f-7      (platform (p_name P1) (p_space S01) (p_state empty))
==> f-8      (platform (p_name P2) (p_space S03) (p_state empty))
==> f-9      (platform (p_name P3) (p_space S04) (p_state empty))
==> f-10     (platform (p_name P4) (p_space S05) (p_state empty))
==> f-11     (platform (p_name P5) (p_space S06) (p_state empty))
==> f-12     (east S01 S02)
==> f-13     (east S02 S03)
==> f-14     (east S04 S05)
==> f-15     (east S05 S06)
==> f-16     (north S01 S04)
==> f-17     (north S02 S05)
==> f-18     (north S03 S06)
==> f-19     (car (c_plate 101) (c_state waiting))
==> f-20     (car (c_plate 102) (c_state waiting))
==> f-21     (car (c_plate 103) (c_state waiting))
==> f-22     (capacity 5)
CLIPS> (run)
FIRE      1 begin: f-0
Parking Begins !

FIRE      2 move_platform_west: f-13, f-3, f-2, f-8
<== f-3      (space (s_name S03) (s_floor 1) (s_state P))
==> f-23     (space (s_name S03) (s_floor 1) (s_state F))
<== f-2      (space (s_name S02) (s_floor 1) (s_state F))
==> f-24     (space (s_name S02) (s_floor 1) (s_state P))
<== f-8      (platform (p_name P2) (p_space S03) (p_state empty))
==> f-25     (platform (p_name P2) (p_space S02) (p_state empty))
platform P2 moved from S03 to S02
FIRE      3 enter_parking: f-21, f-25
<== f-21     (car (c_plate 103) (c_state waiting))
==> f-26     (car (c_plate 103) (c_state P2))
<== f-25     (platform (p_name P2) (p_space S02) (p_state empty))
==> f-27     (platform (p_name P2) (p_space S02) (p_state full))
Car with plate: 103 parked on platform P2
FIRE      4 move_platform_east: f-13, f-24, f-23, f-27
```

```
<== f-24      (space (s_name S02) (s_floor 1) (s_state P))
==> f-28      (space (s_name S02) (s_floor 1) (s_state F))
<== f-23      (space (s_name S03) (s_floor 1) (s_state F))
==> f-29      (space (s_name S03) (s_floor 1) (s_state P))
<== f-27      (platform (p_name P2) (p_space S02) (p_state full))
==> f-30      (platform (p_name P2) (p_space S03) (p_state full))
platform P2 moved from S02 to S03
FIRE      5 move_platform_west: f-13,f-29,f-28,f-30
<== f-29      (space (s_name S03) (s_floor 1) (s_state P))
==> f-31      (space (s_name S03) (s_floor 1) (s_state F))
<== f-28      (space (s_name S02) (s_floor 1) (s_state F))
==> f-32      (space (s_name S02) (s_floor 1) (s_state P))
<== f-30      (platform (p_name P2) (p_space S03) (p_state full))
==> f-33      (platform (p_name P2) (p_space S02) (p_state full))
platform P2 moved from S03 to S02
FIRE      6 move_platform_east: f-13,f-32,f-31,f-33
<== f-32      (space (s_name S02) (s_floor 1) (s_state P))
==> f-34      (space (s_name S02) (s_floor 1) (s_state F))
<== f-31      (space (s_name S03) (s_floor 1) (s_state F))
==> f-35      (space (s_name S03) (s_floor 1) (s_state P))
<== f-33      (platform (p_name P2) (p_space S02) (p_state full))
==> f-36      (platform (p_name P2) (p_space S03) (p_state full))
platform P2 moved from S02 to S03
FIRE      7 move_platform_south: f-17,f-5,f-34,f-10
<== f-5       (space (s_name S05) (s_floor 1) (s_state P))
==> f-37      (space (s_name S05) (s_floor 1) (s_state F))
<== f-34      (space (s_name S02) (s_floor 1) (s_state F))
==> f-38      (space (s_name S02) (s_floor 1) (s_state P))
<== f-10      (platform (p_name P4) (p_space S05) (p_state empty))
==> f-39      (platform (p_name P4) (p_space S02) (p_state empty))
platform P4 moved from S05 to S02
FIRE      8 enter_parking: f-20,f-39
<== f-20      (car (c_plate 102) (c_state waiting))
==> f-40      (car (c_plate 102) (c_state P4))
<== f-39      (platform (p_name P4) (p_space S02) (p_state empty))
==> f-41      (platform (p_name P4) (p_space S02) (p_state full))
Car with plate: 102 parked on platform P4
FIRE      9 move_platform_west: f-15,f-6,f-37,f-11
<== f-6       (space (s_name S06) (s_floor 1) (s_state P))
==> f-42      (space (s_name S06) (s_floor 1) (s_state F))
<== f-37      (space (s_name S05) (s_floor 1) (s_state F))
==> f-43      (space (s_name S05) (s_floor 1) (s_state P))
<== f-11      (platform (p_name P5) (p_space S06) (p_state empty))
==> f-44      (platform (p_name P5) (p_space S05) (p_state empty))
platform P5 moved from S06 to S05
FIRE      10 move_platform_north: f-18,f-35,f-42,f-36
<== f-35      (space (s_name S03) (s_floor 1) (s_state P))
==> f-45      (space (s_name S03) (s_floor 1) (s_state F))
<== f-42      (space (s_name S06) (s_floor 1) (s_state F))
==> f-46      (space (s_name S06) (s_floor 1) (s_state P))
<== f-36      (platform (p_name P2) (p_space S03) (p_state full))
==> f-47      (platform (p_name P2) (p_space S06) (p_state full))
```


platform P2 moved from S03 to S06

```
FIRE 11 move_platform_south: f-18,f-46,f-45,f-47
<== f-46      (space (s_name S06) (s_floor 1) (s_state P))
==> f-48      (space (s_name S06) (s_floor 1) (s_state F))
<== f-45      (space (s_name S03) (s_floor 1) (s_state F))
==> f-49      (space (s_name S03) (s_floor 1) (s_state P))
<== f-47      (platform (p_name P2) (p_space S06) (p_state full))
==> f-50      (platform (p_name P2) (p_space S03) (p_state full))
```

platform P2 moved from S06 to S03

```
FIRE 12 move_platform_east: f-15,f-43,f-48,f-44
<== f-43      (space (s_name S05) (s_floor 1) (s_state P))
==> f-51      (space (s_name S05) (s_floor 1) (s_state F))
<== f-48      (space (s_name S06) (s_floor 1) (s_state F))
==> f-52      (space (s_name S06) (s_floor 1) (s_state P))
<== f-44      (platform (p_name P5) (p_space S05) (p_state empty))
==> f-53      (platform (p_name P5) (p_space S06) (p_state empty))
```

platform P5 moved from S05 to S06

```
FIRE 13 move_platform_west: f-15,f-52,f-51,f-53
<== f-52      (space (s_name S06) (s_floor 1) (s_state P))
==> f-54      (space (s_name S06) (s_floor 1) (s_state F))
<== f-51      (space (s_name S05) (s_floor 1) (s_state F))
==> f-55      (space (s_name S05) (s_floor 1) (s_state P))
<== f-53      (platform (p_name P5) (p_space S06) (p_state empty))
==> f-56      (platform (p_name P5) (p_space S05) (p_state empty))
```

platform P5 moved from S06 to S05

```
FIRE 14 move_platform_north: f-18,f-49,f-54,f-50
<== f-49      (space (s_name S03) (s_floor 1) (s_state P))
==> f-57      (space (s_name S03) (s_floor 1) (s_state F))
<== f-54      (space (s_name S06) (s_floor 1) (s_state F))
==> f-58      (space (s_name S06) (s_floor 1) (s_state P))
<== f-50      (platform (p_name P2) (p_space S03) (p_state full))
==> f-59      (platform (p_name P2) (p_space S06) (p_state full))
```

platform P2 moved from S03 to S06

```
FIRE 15 move_platform_south: f-18,f-58,f-57,f-59
<== f-58      (space (s_name S06) (s_floor 1) (s_state P))
==> f-60      (space (s_name S06) (s_floor 1) (s_state F))
<== f-57      (space (s_name S03) (s_floor 1) (s_state F))
==> f-61      (space (s_name S03) (s_floor 1) (s_state P))
<== f-59      (platform (p_name P2) (p_space S06) (p_state full))
==> f-62      (platform (p_name P2) (p_space S03) (p_state full))
```

platform P2 moved from S06 to S03

```
FIRE 16 move_platform_east: f-15,f-55,f-60,f-56
<== f-55      (space (s_name S05) (s_floor 1) (s_state P))
==> f-63      (space (s_name S05) (s_floor 1) (s_state F))
<== f-60      (space (s_name S06) (s_floor 1) (s_state F))
==> f-64      (space (s_name S06) (s_floor 1) (s_state P))
<== f-56      (platform (p_name P5) (p_space S05) (p_state empty))
==> f-65      (platform (p_name P5) (p_space S06) (p_state empty))
```

platform P5 moved from S05 to S06

```
FIRE 17 move_platform_north: f-17,f-38,f-63,f-41
<== f-38      (space (s_name S02) (s_floor 1) (s_state P))
==> f-66      (space (s_name S02) (s_floor 1) (s_state F))
```

```
<== f-63      (space (s_name S05) (s_floor 1) (s_state F))
==> f-67      (space (s_name S05) (s_floor 1) (s_state P))
<== f-41      (platform (p_name P4) (p_space S02) (p_state full))
==> f-68      (platform (p_name P4) (p_space S05) (p_state full))
platform P4 moved from S02 to S05
FIRE  18 move_platform_west: f-13,f-61,f-66,f-62
<== f-61      (space (s_name S03) (s_floor 1) (s_state P))
==> f-69      (space (s_name S03) (s_floor 1) (s_state F))
<== f-66      (space (s_name S02) (s_floor 1) (s_state F))
==> f-70      (space (s_name S02) (s_floor 1) (s_state P))
<== f-62      (platform (p_name P2) (p_space S03) (p_state full))
==> f-71      (platform (p_name P2) (p_space S02) (p_state full))
platform P2 moved from S03 to S02
FIRE  19 move_platform_south: f-18,f-64,f-69,f-65
<== f-64      (space (s_name S06) (s_floor 1) (s_state P))
==> f-72      (space (s_name S06) (s_floor 1) (s_state F))
<== f-69      (space (s_name S03) (s_floor 1) (s_state F))
==> f-73      (space (s_name S03) (s_floor 1) (s_state P))
<== f-65      (platform (p_name P5) (p_space S06) (p_state empty))
==> f-74      (platform (p_name P5) (p_space S03) (p_state empty))
platform P5 moved from S06 to S03
FIRE  20 move_platform_east: f-15,f-67,f-72,f-68
<== f-67      (space (s_name S05) (s_floor 1) (s_state P))
==> f-75      (space (s_name S05) (s_floor 1) (s_state F))
<== f-72      (space (s_name S06) (s_floor 1) (s_state F))
==> f-76      (space (s_name S06) (s_floor 1) (s_state P))
<== f-68      (platform (p_name P4) (p_space S05) (p_state full))
==> f-77      (platform (p_name P4) (p_space S06) (p_state full))
platform P4 moved from S05 to S06
FIRE  21 move_platform_west: f-15,f-76,f-75,f-77
<== f-76      (space (s_name S06) (s_floor 1) (s_state P))
==> f-78      (space (s_name S06) (s_floor 1) (s_state F))
<== f-75      (space (s_name S05) (s_floor 1) (s_state F))
==> f-79      (space (s_name S05) (s_floor 1) (s_state P))
<== f-77      (platform (p_name P4) (p_space S06) (p_state full))
==> f-80      (platform (p_name P4) (p_space S05) (p_state full))
platform P4 moved from S06 to S05
FIRE  22 move_platform_north: f-18,f-73,f-78,f-74
<== f-73      (space (s_name S03) (s_floor 1) (s_state P))
==> f-81      (space (s_name S03) (s_floor 1) (s_state F))
<== f-78      (space (s_name S06) (s_floor 1) (s_state F))
==> f-82      (space (s_name S06) (s_floor 1) (s_state P))
<== f-74      (platform (p_name P5) (p_space S03) (p_state empty))
==> f-83      (platform (p_name P5) (p_space S06) (p_state empty))
platform P5 moved from S03 to S06
FIRE  23 move_platform_east: f-13,f-70,f-81,f-71
<== f-70      (space (s_name S02) (s_floor 1) (s_state P))
==> f-84      (space (s_name S02) (s_floor 1) (s_state F))
<== f-81      (space (s_name S03) (s_floor 1) (s_state F))
==> f-85      (space (s_name S03) (s_floor 1) (s_state P))
<== f-71      (platform (p_name P2) (p_space S02) (p_state full))
==> f-86      (platform (p_name P2) (p_space S03) (p_state full))
```

platform P2 moved from S02 to S03

```
FIRE 24 move_platform_west: f-13,f-85,f-84,f-86
<== f-85 (space (s_name S03) (s_floor 1) (s_state P))
==> f-87 (space (s_name S03) (s_floor 1) (s_state F))
<== f-84 (space (s_name S02) (s_floor 1) (s_state F))
==> f-88 (space (s_name S02) (s_floor 1) (s_state P))
<== f-86 (platform (p_name P2) (p_space S03) (p_state full))
==> f-89 (platform (p_name P2) (p_space S02) (p_state full))
```

platform P2 moved from S03 to S02

```
FIRE 25 move_platform_east: f-13,f-88,f-87,f-89
<== f-88 (space (s_name S02) (s_floor 1) (s_state P))
==> f-90 (space (s_name S02) (s_floor 1) (s_state F))
<== f-87 (space (s_name S03) (s_floor 1) (s_state F))
==> f-91 (space (s_name S03) (s_floor 1) (s_state P))
<== f-89 (platform (p_name P2) (p_space S02) (p_state full))
==> f-92 (platform (p_name P2) (p_space S03) (p_state full))
```

platform P2 moved from S02 to S03

```
FIRE 26 move_platform_east: f-12,f-1,f-90,f-7
<== f-1 (space (s_name S01) (s_floor 1) (s_state P))
==> f-93 (space (s_name S01) (s_floor 1) (s_state F))
<== f-90 (space (s_name S02) (s_floor 1) (s_state F))
==> f-94 (space (s_name S02) (s_floor 1) (s_state P))
<== f-7 (platform (p_name P1) (p_space S01) (p_state empty))
==> f-95 (platform (p_name P1) (p_space S02) (p_state empty))
```

platform P1 moved from S01 to S02

```
FIRE 27 enter_parking: f-19,f-95
<== f-19 (car (c_plate 101) (c_state waiting))
==> f-96 (car (c_plate 101) (c_state P1))
<== f-95 (platform (p_name P1) (p_space S02) (p_state empty))
==> f-97 (platform (p_name P1) (p_space S02) (p_state full))
```

Car with plate: 101 parked on platform P1

```
FIRE 28 goal: f-96
```

Parking is full

[PRCCODE4] Execution halted during the actions of defrule goal.

CLIPS>

Στρατηγική Simplicity

Παρακάτω βλέπουμε ενδεικτική εκτέλεση με τρία αυτοκίνητα και simplicity στρατηγική.

```
CLIPS> (run)
FIRE    1 begin: f-0
Parking Begins !

FIRE    2 move_platform_east: f-12,f-1,f-2,f-7
<== f-1    (space (s_name S01) (s_floor 1) (s_state P))
==> f-23    (space (s_name S01) (s_floor 1) (s_state F))
<== f-2    (space (s_name S02) (s_floor 1) (s_state F))
==> f-24    (space (s_name S02) (s_floor 1) (s_state P))
<== f-7    (platform (p_name P1) (p_space S01) (p_state empty))
==> f-25    (platform (p_name P1) (p_space S02) (p_state empty))
platform P1 moved from S01 to S02
FIRE    3 enter_parking: f-21,f-25
<== f-21    (car (c_plate 103) (c_state waiting))
==> f-26    (car (c_plate 103) (c_state P1))
<== f-25    (platform (p_name P1) (p_space S02) (p_state empty))
==> f-27    (platform (p_name P1) (p_space S02) (p_state full))
Car with plate: 103 parked on platform P1
FIRE    4 move_platform_south: f-16,f-4,f-23,f-9
<== f-4    (space (s_name S04) (s_floor 1) (s_state P))
==> f-28    (space (s_name S04) (s_floor 1) (s_state F))
<== f-23    (space (s_name S01) (s_floor 1) (s_state F))
==> f-29    (space (s_name S01) (s_floor 1) (s_state P))
<== f-9    (platform (p_name P3) (p_space S04) (p_state empty))
==> f-30    (platform (p_name P3) (p_space S01) (p_state empty))
platform P3 moved from S04 to S01
FIRE    5 move_platform_west: f-14,f-5,f-28,f-10
<== f-5    (space (s_name S05) (s_floor 1) (s_state P))
==> f-31    (space (s_name S05) (s_floor 1) (s_state F))
<== f-28    (space (s_name S04) (s_floor 1) (s_state F))
==> f-32    (space (s_name S04) (s_floor 1) (s_state P))
<== f-10    (platform (p_name P4) (p_space S05) (p_state empty))
==> f-33    (platform (p_name P4) (p_space S04) (p_state empty))
platform P4 moved from S05 to S04
FIRE    6 move_platform_west: f-15,f-6,f-31,f-11
<== f-6    (space (s_name S06) (s_floor 1) (s_state P))
==> f-34    (space (s_name S06) (s_floor 1) (s_state F))
<== f-31    (space (s_name S05) (s_floor 1) (s_state F))
==> f-35    (space (s_name S05) (s_floor 1) (s_state P))
<== f-11    (platform (p_name P5) (p_space S06) (p_state empty))
==> f-36    (platform (p_name P5) (p_space S05) (p_state empty))
platform P5 moved from S06 to S05
FIRE    7 move_platform_north: f-18,f-3,f-34,f-8
<== f-3    (space (s_name S03) (s_floor 1) (s_state P))
==> f-37    (space (s_name S03) (s_floor 1) (s_state F))
<== f-34    (space (s_name S06) (s_floor 1) (s_state F))
==> f-38    (space (s_name S06) (s_floor 1) (s_state P))
```

```
<== f-8      (platform (p_name P2) (p_space S03) (p_state empty))
==> f-39      (platform (p_name P2) (p_space S06) (p_state empty))
platform P2 moved from S03 to S06
FIRE      8 move_platform_east: f-13,f-24,f-37,f-27
<== f-24      (space (s_name S02) (s_floor 1) (s_state P))
==> f-40      (space (s_name S02) (s_floor 1) (s_state F))
<== f-37      (space (s_name S03) (s_floor 1) (s_state F))
==> f-41      (space (s_name S03) (s_floor 1) (s_state P))
<== f-27      (platform (p_name P1) (p_space S02) (p_state full))
==> f-42      (platform (p_name P1) (p_space S03) (p_state full))
platform P1 moved from S02 to S03
FIRE      9 move_platform_south: f-17,f-35,f-40,f-36
<== f-35      (space (s_name S05) (s_floor 1) (s_state P))
==> f-43      (space (s_name S05) (s_floor 1) (s_state F))
<== f-40      (space (s_name S02) (s_floor 1) (s_state F))
==> f-44      (space (s_name S02) (s_floor 1) (s_state P))
<== f-36      (platform (p_name P5) (p_space S05) (p_state empty))
==> f-45      (platform (p_name P5) (p_space S02) (p_state empty))
platform P5 moved from S05 to S02
FIRE     10 enter_parking: f-20,f-45
<== f-20      (car (c_plate 102) (c_state waiting))
==> f-46      (car (c_plate 102) (c_state P5))
<== f-45      (platform (p_name P5) (p_space S02) (p_state empty))
==> f-47      (platform (p_name P5) (p_space S02) (p_state full))
Car with plate: 102 parked on platform P5
FIRE     11 move_platform_west: f-15,f-38,f-43,f-39
<== f-38      (space (s_name S06) (s_floor 1) (s_state P))
==> f-48      (space (s_name S06) (s_floor 1) (s_state F))
<== f-43      (space (s_name S05) (s_floor 1) (s_state F))
==> f-49      (space (s_name S05) (s_floor 1) (s_state P))
<== f-39      (platform (p_name P2) (p_space S06) (p_state empty))
==> f-50      (platform (p_name P2) (p_space S05) (p_state empty))
platform P2 moved from S06 to S05
FIRE     12 move_platform_north: f-18,f-41,f-48,f-42
<== f-41      (space (s_name S03) (s_floor 1) (s_state P))
==> f-51      (space (s_name S03) (s_floor 1) (s_state F))
<== f-48      (space (s_name S06) (s_floor 1) (s_state F))
==> f-52      (space (s_name S06) (s_floor 1) (s_state P))
<== f-42      (platform (p_name P1) (p_space S03) (p_state full))
==> f-53      (platform (p_name P1) (p_space S06) (p_state full))
platform P1 moved from S03 to S06
FIRE     13 move_platform_east: f-13,f-44,f-51,f-47
<== f-44      (space (s_name S02) (s_floor 1) (s_state P))
==> f-54      (space (s_name S02) (s_floor 1) (s_state F))
<== f-51      (space (s_name S03) (s_floor 1) (s_state F))
==> f-55      (space (s_name S03) (s_floor 1) (s_state P))
<== f-47      (platform (p_name P5) (p_space S02) (p_state full))
==> f-56      (platform (p_name P5) (p_space S03) (p_state full))
platform P5 moved from S02 to S03
FIRE     14 move_platform_south: f-17,f-49,f-54,f-50
<== f-49      (space (s_name S05) (s_floor 1) (s_state P))
==> f-57      (space (s_name S05) (s_floor 1) (s_state F))
```

```
<== f-54      (space (s_name S02) (s_floor 1) (s_state F))
==> f-58      (space (s_name S02) (s_floor 1) (s_state P))
<== f-50      (platform (p_name P2) (p_space S05) (p_state empty))
==> f-59      (platform (p_name P2) (p_space S02) (p_state empty))
platform P2 moved from S05 to S02
FIRE  15 enter_parking: f-19,f-59
<== f-19      (car (c_plate 101) (c_state waiting))
==> f-60      (car (c_plate 101) (c_state P2))
<== f-59      (platform (p_name P2) (p_space S02) (p_state empty))
==> f-61      (platform (p_name P2) (p_space S02) (p_state full))
Car with plate: 101 parked on platform P2
FIRE  16 goal: f-60
Parking is full
[PRCCODE4] Execution halted during the actions of defrule goal.
CLIPS>
```

Στρατηγική Complexity

Παρακάτω βλέπουμε ενδεικτική εκτέλεση με τρία αυτοκίνητα και complexity στρατηγική. (φαίνεται ίδιο με το simplicity)

```
CLIPS> (run)
FIRE    1 begin: f-0
Parking Begins !

FIRE    2 move_platform_east: f-12,f-1,f-2,f-7
<== f-1      (space (s_name S01) (s_floor 1) (s_state P))
==> f-23     (space (s_name S01) (s_floor 1) (s_state F))
<== f-2      (space (s_name S02) (s_floor 1) (s_state F))
==> f-24     (space (s_name S02) (s_floor 1) (s_state P))
<== f-7      (platform (p_name P1) (p_space S01) (p_state empty))
==> f-25     (platform (p_name P1) (p_space S02) (p_state empty))
platform P1 moved from S01 to S02
FIRE    3 enter_parking: f-21,f-25
<== f-21     (car (c_plate 103) (c_state waiting))
==> f-26     (car (c_plate 103) (c_state P1))
<== f-25     (platform (p_name P1) (p_space S02) (p_state empty))
==> f-27     (platform (p_name P1) (p_space S02) (p_state full))
Car with plate: 103 parked on platform P1
FIRE    4 move_platform_south: f-16,f-4,f-23,f-9
<== f-4      (space (s_name S04) (s_floor 1) (s_state P))
==> f-28     (space (s_name S04) (s_floor 1) (s_state F))
<== f-23     (space (s_name S01) (s_floor 1) (s_state F))
==> f-29     (space (s_name S01) (s_floor 1) (s_state P))
<== f-9      (platform (p_name P3) (p_space S04) (p_state empty))
==> f-30     (platform (p_name P3) (p_space S01) (p_state empty))
platform P3 moved from S04 to S01
FIRE    5 move_platform_west: f-14,f-5,f-28,f-10
<== f-5      (space (s_name S05) (s_floor 1) (s_state P))
==> f-31     (space (s_name S05) (s_floor 1) (s_state F))
<== f-28     (space (s_name S04) (s_floor 1) (s_state F))
==> f-32     (space (s_name S04) (s_floor 1) (s_state P))
<== f-10     (platform (p_name P4) (p_space S05) (p_state empty))
==> f-33     (platform (p_name P4) (p_space S04) (p_state empty))
platform P4 moved from S05 to S04
FIRE    6 move_platform_west: f-15,f-6,f-31,f-11
<== f-6      (space (s_name S06) (s_floor 1) (s_state P))
==> f-34     (space (s_name S06) (s_floor 1) (s_state F))
<== f-31     (space (s_name S05) (s_floor 1) (s_state F))
==> f-35     (space (s_name S05) (s_floor 1) (s_state P))
<== f-11     (platform (p_name P5) (p_space S06) (p_state empty))
==> f-36     (platform (p_name P5) (p_space S05) (p_state empty))
platform P5 moved from S06 to S05
FIRE    7 move_platform_north: f-18,f-3,f-34,f-8
<== f-3      (space (s_name S03) (s_floor 1) (s_state P))
==> f-37     (space (s_name S03) (s_floor 1) (s_state F))
<== f-34     (space (s_name S06) (s_floor 1) (s_state F))
```

```
==> f-38      (space (s_name S06) (s_floor 1) (s_state P))
<== f-8       (platform (p_name P2) (p_space S03) (p_state empty))
==> f-39      (platform (p_name P2) (p_space S06) (p_state empty))
platform P2 moved from S03 to S06
FIRE      8 move_platform_east: f-13,f-24,f-37,f-27
<== f-24      (space (s_name S02) (s_floor 1) (s_state P))
==> f-40      (space (s_name S02) (s_floor 1) (s_state F))
<== f-37      (space (s_name S03) (s_floor 1) (s_state F))
==> f-41      (space (s_name S03) (s_floor 1) (s_state P))
<== f-27      (platform (p_name P1) (p_space S02) (p_state full))
==> f-42      (platform (p_name P1) (p_space S03) (p_state full))
platform P1 moved from S02 to S03
FIRE      9 move_platform_south: f-17,f-35,f-40,f-36
<== f-35      (space (s_name S05) (s_floor 1) (s_state P))
==> f-43      (space (s_name S05) (s_floor 1) (s_state F))
<== f-40      (space (s_name S02) (s_floor 1) (s_state F))
==> f-44      (space (s_name S02) (s_floor 1) (s_state P))
<== f-36      (platform (p_name P5) (p_space S05) (p_state empty))
==> f-45      (platform (p_name P5) (p_space S02) (p_state empty))
platform P5 moved from S05 to S02
FIRE     10 enter_parking: f-20,f-45
<== f-20      (car (c_plate 102) (c_state waiting))
==> f-46      (car (c_plate 102) (c_state P5))
<== f-45      (platform (p_name P5) (p_space S02) (p_state empty))
==> f-47      (platform (p_name P5) (p_space S02) (p_state full))
Car with plate: 102 parked on platform P5
FIRE     11 move_platform_west: f-15,f-38,f-43,f-39
<== f-38      (space (s_name S06) (s_floor 1) (s_state P))
==> f-48      (space (s_name S06) (s_floor 1) (s_state F))
<== f-43      (space (s_name S05) (s_floor 1) (s_state F))
==> f-49      (space (s_name S05) (s_floor 1) (s_state P))
<== f-39      (platform (p_name P2) (p_space S06) (p_state empty))
==> f-50      (platform (p_name P2) (p_space S05) (p_state empty))
platform P2 moved from S06 to S05
FIRE     12 move_platform_north: f-18,f-41,f-48,f-42
<== f-41      (space (s_name S03) (s_floor 1) (s_state P))
==> f-51      (space (s_name S03) (s_floor 1) (s_state F))
<== f-48      (space (s_name S06) (s_floor 1) (s_state F))
==> f-52      (space (s_name S06) (s_floor 1) (s_state P))
<== f-42      (platform (p_name P1) (p_space S03) (p_state full))
==> f-53      (platform (p_name P1) (p_space S06) (p_state full))
platform P1 moved from S03 to S06
FIRE     13 move_platform_east: f-13,f-44,f-51,f-47
<== f-44      (space (s_name S02) (s_floor 1) (s_state P))
==> f-54      (space (s_name S02) (s_floor 1) (s_state F))
<== f-51      (space (s_name S03) (s_floor 1) (s_state F))
==> f-55      (space (s_name S03) (s_floor 1) (s_state P))
<== f-47      (platform (p_name P5) (p_space S02) (p_state full))
==> f-56      (platform (p_name P5) (p_space S03) (p_state full))
platform P5 moved from S02 to S03
FIRE     14 move_platform_south: f-17,f-49,f-54,f-50
<== f-49      (space (s_name S05) (s_floor 1) (s_state P))
```



```
==> f-57      (space (s_name S05) (s_floor 1) (s_state F))
<== f-54      (space (s_name S02) (s_floor 1) (s_state F))
==> f-58      (space (s_name S02) (s_floor 1) (s_state P))
<== f-50      (platform (p_name P2) (p_space S05) (p_state empty))
==> f-59      (platform (p_name P2) (p_space S02) (p_state empty))
platform P2 moved from S05 to S02
FIRE 15 enter_parking: f-19,f-59
<== f-19      (car (c_plate 101) (c_state waiting))
==> f-60      (car (c_plate 101) (c_state P2))
<== f-59      (platform (p_name P2) (p_space S02) (p_state empty))
==> f-61      (platform (p_name P2) (p_space S02) (p_state full))
Car with plate: 101 parked on platform P2
FIRE 16 goal: f-60
Parking is full
[PRCCODE4] Execution halted during the actions of defrule goal.
CLIPS>
```

Βιβλιογραφία

Γεωργούλη, Α. (2015). *Τεχνητή Νοημοσύνη - Μια εισαγωγική προσέγγιση*.

<https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/3381/1/Τεχνητή%20Νοημοσύνη.pdf>