



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ**
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Ακαδημαϊκό Έτος 2022-2023

ΤΕΧΝΙΚΗ ΑΝΑΦΟΡΑ

3^η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

<ΠΡΟΒΛΗΜΑ ΤΟΥ ΑΝΕΛΚΥΣΤΗΡΑ>

<ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ ΠΟΤΑΜΙΑΝΟΣ>

A.M.: <1084537>

Πάτρα 2022-23

(α) Τα κατάλληλα πέντε κατηγορήματα προκειμένου να μπορέσουν να εκφραστούν τα γεγονότα που αντικατοπτρίζουν κάθε κατάσταση του προβλήματος είναι τα εξής:

- X: Επιβάτες, Y: Όροφος
- **at(X, Y):** Το συγκεκριμένο κατηγορήμα είναι αληθές αν ο ανελκυστήρας βρίσκεται στον όροφο «Y» και «X» επιβάτες βρίσκονται μέσα του.
- **waiting(X, Y):** Το συγκεκριμένο κατηγορήμα είναι αληθές εφόσον υπάρχουν «X» επιβάτες που περιμένουν στον όροφο «Y».
- **destination(X, Y):** Το συγκεκριμένο κατηγορήμα είναι αληθές αν ένας επιβάτης έχει τον όροφο «Y» ως τελικό του προορισμό.
- **inElevator(X):** Το συγκεκριμένο κατηγορήμα είναι αληθές αν ένας επιβάτης «X» βρίσκεται αυτή την στιγμή μέσα στον ανελκυστήρα.
- **called(X):** Το συγκεκριμένο κατηγορήμα είναι αληθές αν κάλεσαν τον ανελκυστήρα από τον όροφο «X».

Τα παρακάτω παραδείγματα θα αποσαφηνίσουν πλήρως τα παραπάνω κατηγορήματα.

(1) Έστω ότι ο ανελκυστήρας βρίσκεται αυτή την στιγμή στο ισόγειο και επιβαίνουν σε αυτόν δύο άτομα που έχουν ως τελικό προορισμό τους τον τρίτο όροφο. Αυτό θα αναπαρασταθεί ως εξής:

```
at(2, 0)           // ο ανελκυστήρας βρίσκεται στον όροφο 0 και 2 επιβάτες βρίσκονται μέσα του
destination(1, 3)  // ο επιβάτης «1» έχει τον όροφο 3 ως τελικό του προορισμό.
destination(2, 3)  // ο επιβάτης «2» έχει τον όροφο 3 ως τελικό του προορισμό.
inElevator(1)      // ο επιβάτης «1» βρίσκεται αυτή την στιγμή μέσα στον ανελκυστήρα.
inElevator(2)      // ο επιβάτης «2» βρίσκεται αυτή την στιγμή μέσα στον ανελκυστήρα.
```

(2) Έστω ότι υπάρχει ένα άτομο που περιμένει στον πρώτο όροφο και έχει το ισόγειο ως τελικό προορισμό του και πως ο ανελκυστήρας αυτή την στιγμή βρίσκεται στον τρίτο όροφο με κανέναν επιβάτη μέσα:

```
at(0, 3)           // ο ανελκυστήρας βρίσκεται στον όροφο 3 και 0 επιβάτες βρίσκονται μέσα του
destination(1, 0)  // ο επιβάτης «1» έχει τον όροφο 0 (ισόγειο) ως τελικό του προορισμό.
waiting(1, 1)      // υπάρχει «1» επιβάτης που περιμένει στον όροφο «1».
```

(3) Έστω ότι ο ανελκυστήρας έλαβε κλήση από τον δεύτερο όροφο αλλά εκείνη την στιγμή βρίσκεται στο ισόγειο με έναν επιβαίνοντα ο οποίος έχει τον τρίτο όροφο ως τελικό του προορισμό:

```
at(1, 0)           // ο ανελκυστήρας βρίσκεται στον όροφο 0 και 1 επιβάτες βρίσκονται μέσα του
called(2)          // κάλεσαν τον ανελκυστήρα από τον όροφο «2».
destination(1, 3)  // ο επιβάτης «1» έχει τον όροφο 3 ως τελικό του προορισμό.
inElevator(1)      // ο επιβάτης «1» βρίσκεται αυτή την στιγμή μέσα στον ανελκυστήρα.
```

(β) Ένας πιθανός τρόπος απεικόνισης λαμβάνοντας υπόψιν και τα κατηγορήματα του ερωτήματος (α) είναι ο εξής:

Κατάσταση ανελκυστήρα:

Τρέχον Όροφος: Y

Επιβαίνοντες: X (P1, P2, ..., Px) // P=ονόματα επιβατών

Άτομα που περιμένουν:

```
Floor 0: Z0          // Z0=ονόματα επιβατών που περιμένουν στο ισόγειο
Floor 1: Z1          // Z1=ονόματα επιβατών που περιμένουν στον πρώτο όροφο.
Floor 2: Z2          // Z2=ονόματα επιβατών που περιμένουν στον δεύτερο όροφο.
Floor 3: Z3          // Z3=ονόματα επιβατών που περιμένουν στον τρίτο όροφο.
```

Προορισμός επιβαινόντων:

```
Destination 0: W0    // W0=ονόματα επιβατών που έχουν το ισόγειο ως προορισμό τους.
Destination 1: W1    // W1=ονόματα επιβατών που έχουν το πρώτο ως προορισμό τους.
Destination 2: W2    // W2=ονόματα επιβατών που έχουν το δεύτερο ως προορισμό τους.
Destination 3: W3    // W3=ονόματα επιβατών που έχουν το τρίτο ως προορισμό τους.
```

Με αυτόν τον τρόπο κάνουμε εξαιρετικά εύκολο να απεικονίσουμε τις καταστάσεις του ανελκυστήρα, που θα μας φανεί ακόμα πιο χρήσιμο και εύκολο στα επόμενα ερωτήματα.

Έστω ότι ο ανελκυστήρας μας βρίσκεται στον δεύτερο όροφο και έχει δύο επιβάτες (τους P1 και P2), ταυτόχρονα υπάρχει ένα άτομο που βρίσκεται στο ισόγειο και έχει ως προορισμό του τον τρίτο όροφο (ο P3) και υπάρχουν ακόμα άλλα δύο άτομα που περιμένουν στον πρώτο όροφο και έχουν τον δεύτερο όροφο ως προορισμό τους (έστω οι P4 και P5). Στο συγκεκριμένο πολύ απλό παράδειγμα θα είχαμε την εξής απεικόνιση:

Κατάσταση ανελκυστήρα:
Τρέχον Όροφος: 2
Επιβαίνοντες: 2 (P1, P2)

Άτομα που περιμένουν:
Floor 0: 1 (P3)
Floor 1: 2 (P4, P5)
Floor 2: 0
Floor 3: 0

Προορισμός επιβαινόντων:
Destination 0: 0
Destination 1: 0
Destination 2: 2 (P4, P5)
Destination 3: 1 (P3)

(γ1)

% Καλούμε τον ανελκυστήρα σε συγκεκριμένο όροφο

```
callElevator(Floor) :-  
    called(Floor),  
    write('Ο Anelkistiras exei idi kalestei se auto ton orof').  
callElevator(Floor) :-  
    assert(called(Floor)).
```

% Μετακινούμε τον ανελκυστήρα σε συγκεκριμένο όροφο

```
moveElevator(Floor) :-  
    at(_, Floor), % remove the Passengers variable  
    write('Elevator already at this floor.').  
moveElevator(Floor) :-  
    at(_, CurrentFloor), % remove the Passengers variable  
    retract(at(_, CurrentFloor)), % remove the Passengers variable  
    assert(at(_, Floor)), % remove the Passengers variable  
    write('Ο anelkistiras metakinithike ston orof'), write(Floor).
```

% Μπαινουν επιβάτες μεσα

```
pickUpPassengers(Floor) :-  
    at(Passengers, Floor),  
    Passengers >= 3,  
    write('Το asanser einai gemato').  
pickUpPassengers(Floor) :-  
    at(Passengers, Floor),  
    waiting(Waiting, Floor),  
    Passengers + Waiting < 3,  
    retract(waiting(Waiting, Floor)),  
    NewPassengers is Passengers + Waiting,  
    assert(at(NewPassengers, Floor)),  
    write('Μpikan'), write(Waiting), write(' epivates ston orof '), write(Floor).
```

% Κατεβαίνουν επιβάτες στον προορισμό τους

```
dropOffPassengers(Floor) :-  
    at(Passengers, Floor),  
    Passengers = 0,  
    write('Το asanser einai adeio').  
dropOffPassengers(Floor) :-  
    at(Passengers, Floor),  
    Passengers > 0,  
    findall(Passenger, (inElevator(Passenger), destination(Passenger, Floor)), PassengersToDropOff),  
    length(PassengersToDropOff, NumToDropOff),  
    retract(at(Passengers, Floor)),  
    NewPassengers is Passengers - NumToDropOff,  
    assert(at(NewPassengers, Floor)),  
    maplist(retract, PassengersToDropOff),  
    write('katevikan'), write(NumToDropOff), write(' epivates ston orof '), write(Floor).
```