



ΕΝΟΤΗΤΑ 1.2

Εισαγωγή στις Βασικές Έννοιες Δραστών

Διδάσκων Ενότητας:
Κώστας Κοντογιάννης

ΜΕΡΟΣ 1: Επίλυση Προβλημάτων στο Χώρο των καταστάσεων

Περίληψη Ενότητας 1.2



2

Θέματα

- Βασικοί Τύποι Δραστών

Υλικό

- Διαφάνειες παρουσίασης
- Περίληψη παρουσίασης
- Κεφάλαιο 1 Βιβλίου

Βιβλιογραφία - Αναφορές



3

- Υλικό για αυτή τη διάλεξη συγκεντρώθηκε από τις συνοδευτικές παρουσιάσεις του βιβλίου, και από το βιβλίο Artificial Intelligence [Russell].

Γενικά Περί Δραστών (Agents)



4

- Με το όρο δράστη (agent) στη περιοχή της Τεχνητής Νοημοσύνης, περιγράφουμε ένα σύστημα (π.χ. Πρόγραμμα) το οποίο λαμβάνει μέσω αισθητήρων (sensors) ή κάποιας άλλης διαπροσωπείας ερεθίσματα από το περιβάλλον του και στη συνέχεια αντιδρά σε αυτά τα ερεθίσματα με κάποιες ενέργειες μέσω κατάλληλων μηχανισμών (effectors)
- Ένας Ορθολογικός Δράστης (rational agent) είναι ένας δράστης που πάντα αντιδρά με το σωστό τρόπο στα ερεθίσματα που λαμβάνει από το περιβάλλον του (does the *right thing*)
- Το πρόβλημα λοιπόν είναι να
 - ▣ Μοντελοποιήσουμε τις ενέργειες του δράστη σαν συνάρτηση των ερεθισμάτων που λαμβάνει (και πιθανόν και σαν συνάρτηση άλλων παραγόντων όπως η κατάσταση που βρίσκεται ο δράστης κλπ.)
 - ▣ Αναπτύξουμε τεχνικές με τις οποίες θα μπορέσουμε να προσδιορίσουμε πότε και πως μετράμε την απόδοση του δράστη
- Συνεπώς ένας Ορθολογικός Δράστης για τη λειτουργία του βασίζεται στα:
 - ▣ Σε κάποια μετρική / μετρικές που ορίζουν το βαθμό απόδοσης του δράστη
 - ▣ Την ακολουθία ερεθισμάτων που έχει δει ο δράστης μέχρι κάποια χρονική στιγμή (percept sequence)
 - ▣ Τη γνώση που έχει ο δράστης για το περιβάλλον του
 - ▣ Τις ενέργειες που μπορεί ο δράστης να εκτελέσει σε κάθε πιθανή κατάσταση και ερέθισμα

Ιδανικοί Ορθολογικοί Δράστες



5

- Η δομή ενός ορθολογικού δράστη είναι
Ορθολογικός δράστης = Αρχιτεκτονική + Πρόγραμμα
- Αρχιτεκτονική
 - ▣ Επιτρέπει τα ερεθίσματα να διοχετεύονται στο Πρόγραμμα
 - ▣ Εκτελεί το Πρόγραμμα
 - ▣ Διοχετεύει τα αποτελέσματα του Προγράμματος σε μηχανισμούς δράσης (actuators, effectors)
- Πρόγραμμα
 - ▣ Ορίζει την απεικόνιση (ακολουθία ερεθισμάτων → ενέργεια)
 - ▣ Ανάλογα με την απεικόνιση έχουμε Table Driven Agents, Simple Reflex Agents, Reflex Agents with Internal State, Goal-based Agents, Utility-based Agents
- Ένας Ιδανικός Ορθολογικός Δράστης (Ideal Rational Agent) είναι ο δράστης ο οποίος για κάθε πιθανή κατάσταση και ερεθίσματα έχει τη δυνατότητα και γνώση να ενεργήσει με κάποιο τρόπο που θα μεγιστοποιήσει την επίδοσή του

Κατηγοριοποίηση Δραστών



6

- Ένας δράστης (agent) λοιπόν περιγράφεται σε γενικές γραμμές από την αρχιτεκτονική του και τη συνάρτηση ενεργειών δράσης (agent activity function)
- Ανάλογα με το πώς δομείται ένας δράστης κατηγοριοποιείται ως:
 - ▣ Table Driven Agent,
 - ▣ Simple Reflex Agent,
 - ▣ Reflex Agent with Internal State,
 - ▣ Goal-based Agent,
 - ▣ Utility-based Agent

Table Driven Agents



7

- Οι Table Driven Agents χρησιμοποιούν πίνακες για να αντιστοιχήσουν τα ερεθίσματα που λαμβάνει ο δράστης από το περιβάλλον και τη κατάσταση του δράστη με τις ενέργειες που πρέπει να εκτελεστούν
- Αυτή η προσέγγιση έχει αρκετά μειονεκτήματα και πιο συγκεκριμένα:
 - Είναι απαραίτητο να κρατιέται στη μνήμη ολόκληρη η ακολουθία ερεθισμάτων (percept sequence)
 - Είναι δύσκολο να οριστεί ένας πλήρης πίνακας αντιστοίχισης
 - Ο Δράστης δεν μπορεί να παρουσιάσει αυτόνομη συμπεριφορά

Simple Reflex Agents



8

- Η βασική ιδέα πίσω από τους Simple Reflex Agents είναι η αναγνώριση / αποθήκευση συχνών και επαναλαμβανόμενων σχέσεων μεταξύ ερεθισμάτων και ενεργειών
- Αυτές οι κοινές και επαναλαμβανόμενες σχέσεις μοντελοποιούνται με τη μορφή κανόνων που ονομάζονται κανόνες συνθήκης-ενέργειας (condition-action rules)
- Ένας Simple Reflex Agent δουλεύει με την αρχή όπου το πρόγραμμα επιλέγει τον κατάλληλο κανόνα, και στη συνέχεια εφαρμόζει την αντίστοιχη ενέργεια
- Ο ψευδοκώδικας που περιγράφει ένα Simple Reflex Agent είναι

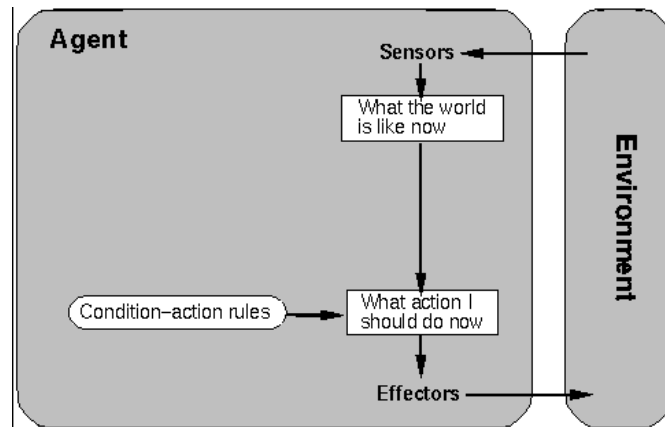

```

Function Simple-Reflex-Agent(percept)
  static: rules, /* condition-action rules */
  state ← Intercept_input(percept)
  rule ← Rule_match(state, rules)
  action ← Rule_Action(rule)
  return(action)
      
```

Simple Reflex Agent - Σχηματικά



9



Δράστης με Εσωτερική Κατάσταση – Reflex Agent with Internal State



10

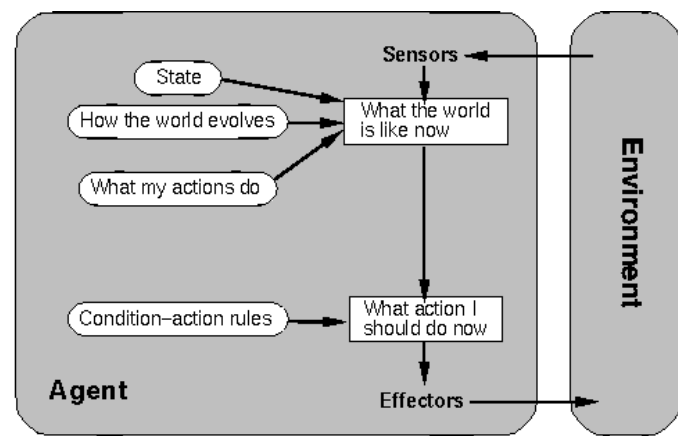
- Τα ερεθίσματα που λαμβάνει ένας δράστης μέσω των αισθητήρων του, συχνά δεν δίνουν την συνολική εικόνα του περιβάλλοντος
- Χρειάζεται λοιπόν ο δράστης να έχει μηχανισμούς για να διακρίνει και να ξεχωρίζει ανάμεσα σε διαφορετικές καταστάσεις του περιβάλλοντος που μπορεί να παράγουν τα ίδια ερεθίσματα
- Ο μηχανισμός που χρησιμοποιείται εδώ είναι η διατήρηση ενός μοντέλου εσωτερικής κατάστασης που επιτρέπει την διαφοροποίηση των περιπτώσεων και την επιλογή των δράσεων
- Ο ψευδοκώδικας που προσδιορίζει τη συμπεριφορά ενός Reflex Agent με Εσωτερική Κατάσταση είναι:

```

Function Reflex-Agent-With-State(percept)
    static: state, /* description of the current world state */
           rules // set of condition-action rules //
    state ← Update_State(state, percept)
    rule ← Rule_Match(state, rules)
    action ← Rule_Action(rule)
    state ← Update_State(state, action)
    return(action)
  
```

Δράστης με Εσωτερική Κατάσταση

11



Δράστης Επιδίωξης Στόχου – Goal Based Agent

12

- Η γνώση της κατάστασης του περιβάλλοντος μερικές φορές δεν είναι αρκετή για την επιλογή της σωστής ενέργειας
- Οι Δράστες Επιδίωξης Στόχου βασίζονται στην ιδέα ότι ο δράστης έχει και κάποιους συγκεκριμένους στόχους που πρέπει να επιτύχει (όχι απλά να επιτελέσει μια ενέργεια)
- Η γνώση των στόχων που πρέπει να επιτευχθούν συνδυάζεται με όλες τις άλλες πληροφορίες που έχει στη διάθεσή του ο δράστης ώστε να επιλέξει την κατάλληλη ενέργεια
- Η επιλογή μιας κατάλληλης ενέργειας είναι αποτέλεσμα μιας διαδικασίας έρευνας (search) και αλγόριθμων μεθόδευσης ενεργειών (planning) – Εύρεση μιας ακολουθίας ενεργειών που επιτυγχάνουν το στόχο

Ψευδοκώδικας Δράστη Επιδίωξης Στόχου



13

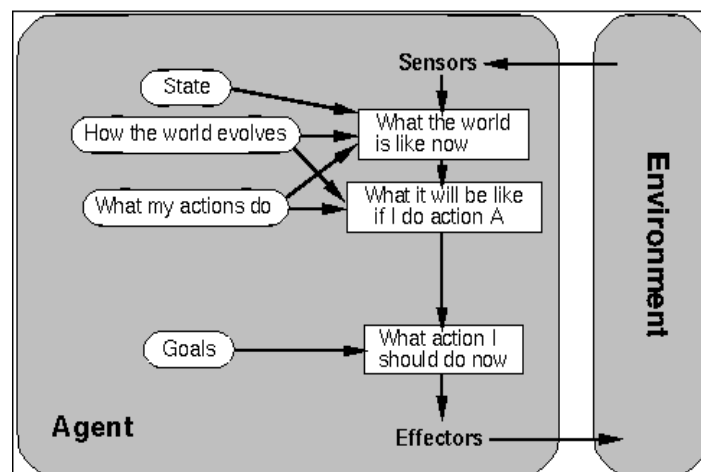
```

Function Goal-Based-Agent(percept)
  static: state, /* description of the current world state */
         rules /* set of condition-action rules */
         goal /* set of specific success states */
  state ← Update_State(state, percept)
  rule ← Rule_Match(state, rules)
  action ← Rule_Action(rule)
  state ← Update_State(state, action)
  if (state in goal) then
    return (action)
  else
    percept ← Obtain_Percept(state, goal)
    return(Goal-Based-Agent(percept))
  
```

Δράστης Επιδίωξης Στόχου - Σχηματικά



14



Δράστης Ωφέλειας – Utility Based Agent



15

- Η συμπεριφορά ενός δράστη να επιτύχει τους στόχους του δεν εγγυάται ότι αυτός ο δράστης εκτελεί και τις καλύτερες δυνατές ενέργειες ή ότι έχει υψηλή πιστότητα
- Στους δράστες επίτευξης στόχου οι καταστάσεις του δράστη χαρακτηρίζονται ως επιτυχείς (τελική επίτευξη στόχου) ή ανεπιτυχείς (ο στόχος δεν έχει ακόμη επιτευχθεί)
- Χρειαζόμαστε λοιπόν μια μέθοδο για να διακρίνουμε πόσο είναι το κέρδος ή το σκορ κάθε κατάστασης σε σχέση με το στόχο
- Χρησιμοποιούμε για αυτό το σκοπό μια συνάρτηση που ονομάζουμε συνάρτηση ωφέλειας (utility function)
- Η συνάρτηση ωφέλειας αποδίδει σε κάθε κατάσταση ένα σκορ και έτσι επιτρέπει
 - ▣ Την επιλογή εναλλακτικών ενεργειών
 - ▣ Την επιλογή κατάλληλων και επιτεύξιμων στόχων

Ψευδοκώδικας Δράστη Ωφέλειας



16

Function Goal-Based-Agent(percept)

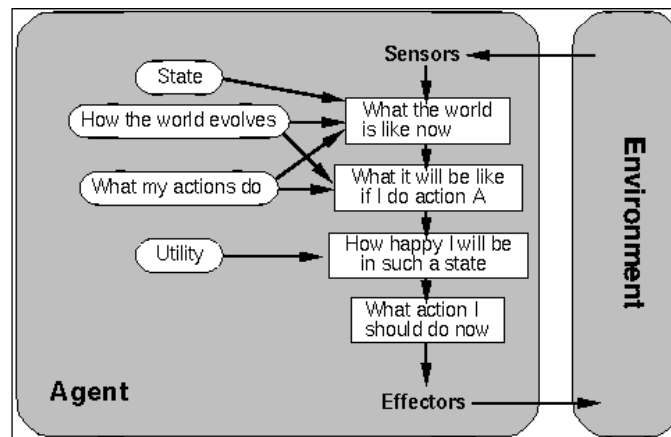
```

static: state, /* description of the current world state */
        rules /* set of condition-action rules */
        goal /* set of specific success states */
state ← Update_State(state, percept)
rule ← Rule_Match(state, rules)
action ← Rule_Action(rule)
state ← Update_State(state, action)
score ← Obtain_Score(state)
if (state in goal) and Best_Score(score) then
    return(action)
else
    percept ← Obtain_Percept(state, goal)
    return(Goal-Based-Agent(percept))
  
```


Δράστες Ωφέλειας – Σχηματικά



17



Ταξινόμηση Τύπων Περιβάλλοντος



18

- Τα περιβάλλοντα στα οποία ενεργούν οι δράστες ταξινομούνται ως:
 - ▣ Προσιτά / Μη-Προσιτά (οι αισθητήρες μπορούν να έχουν πρόσβαση σε όλο ή μόνο σε μέρος του περιβάλλοντος)
 - ▣ Νομοτελειακά / Μη-Νομοτελειακά (η κατάσταση του περιβάλλοντος είναι νομοτελειακά ή μη ορισμένη)
 - ▣ Επεισοδικά / Μη-Επεισοδικά (η συμπεριφορά του δράστη βασίζεται πρώτα στη εισαγωγή στοιχείων και μετά στην επιλογή της ενέργειας acting vs. planning)
 - ▣ Στατικά / Δυναμικά (το περιβάλλον παραμένει στατικό ή αλλάζει καθώς ο δράστης ενεργεί)
 - ▣ Συνεχή / Μη-συνεχή περιβάλλοντα (distinct percepts vs. continuous percept sequences)

Παραδείγματα



19

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Chess with a clock	Yes	Yes	No	Semi	Yes
Chess without a clock	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Medical diagnosis system	No	No	No	No	No
Image-analysis system	Yes	Yes	Yes	Semi	No
Part-picking robot	No	No	Yes	No	No
Refinery controller	No	No	No	No	No
Interactive English tutor	No	No	No	No	Yes