

---

# **Introduzione ai Sistemi a Classificatori**

---

**Pier Luca Lanzi**  
**Dipartimento di Elettronica e Informazione**  
**Politecnico di Milano**  
**Venerdì 21 Marzo 1997**  
[www.elet.polimi.it/people/lanzi](http://www.elet.polimi.it/people/lanzi)

---

## **Sommario**

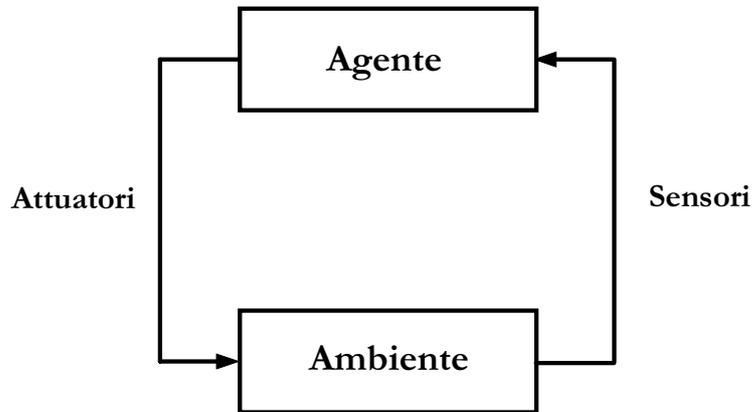
---

- **Agente e Ambiente**
- **Approcci alla soluzione**
- **Sistemi a Classificatori di Wilson (XCS)**
- **Esempi**
- **Sistemi a Classificatori Tradizionali**

## Interazione Agente/Ambiente

---

Un agente è un unità che agisce o opera un un certo ambiente.



## Agente con Sistemi a Classificatori

---

Agente è pilotato utilizzando un insieme di regole di produzione dette classificatori.

### Soluzioni Generali

- Le regole sono scelte a priori: un esperto definisce il comportamento dell'agente esprimendo attraverso delle regole (Sistemi esperti, Controllori a logica fuzzy).
- L'agente apprende le regole migliori per interagire con l'ambiente.

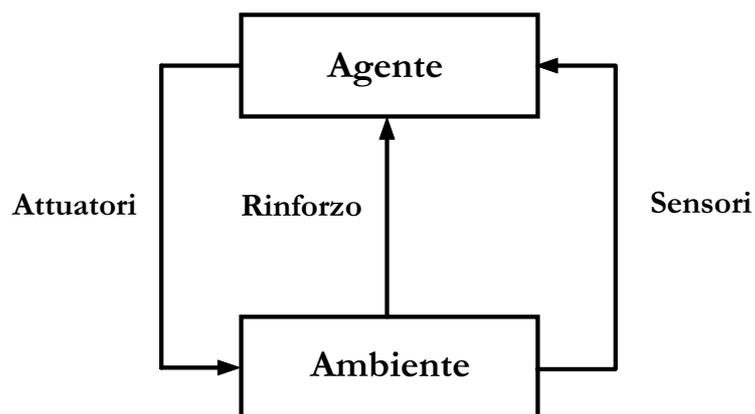
## Approccio con Sistemi a Classificatori

---

# Apprendimento con rinforzo + Sistemi evolutivi

## Apprendimento con Rinforzo

---



Il nostro agente si comporta bene se riceve molti “premi” ovvero ottiene molto rinforzo.

L'agente apprende come deve agire per prendere più rinforzo possibile.

## Apprendimento con Rinforzo e Sistemi a Classificatori

---

### Lo Scopo

- Trovare un insieme di regole che se adoperate permettano di ottenere il maggior rinforzo (premio) possibile.
- La soluzione è rappresentata da una serie di regole da utilizzare in seguito per controllare l'agente.
- Bisogna definire un sistema evolutivo che possa ottenere il tipo di soluzione voluta.
- Due metodi sono possibili: l'approccio Pittsburgh e l'approccio Michigan.

### Approccio Pittsburgh (1)

---

- Utilizza un algoritmo genetico standard.
- Individuo rappresenta un possibile insieme di regole di grandezza fissata.
- Crossover e Mutazione vengono ridefiniti.
- La fitness di un individuo viene calcolata come il rinforzo che un'insieme di regole riesce ad ottenere in  $N$  passi di computazione.

#### Esempio

$n$  regole {if  $a_1$  then  $b_1$ , ..., if  $a_n$  then  $b_n$ }

$a_1$	$b_1$	...	$a_n$	$b_n$
-------	-------	-----	-------	-------

## Approccio Pittsburgh (2)

---

### Algoritmo di Apprendimento

- Si valuta la fitness di tutti gli individui.
  - Le regole sono date all'agente che viene fatto funzionare per un certo tempo.
  - La fitness è uguale a tutto il reward che l'agente è riuscito ad accumulare durante il funzionamento.
- Si applica l'operatore di selezione/riproduzione.
- Si applicano gli operatori di ricombinazione.
- Il processo termina dopo un certo numero di passi, quando gli individui sono sufficientemente buoni.

## Approccio Michigan

---

- Un unico insieme di regole.
- Ogni istante viene selezionata la regola più opportuna.
- Se il sistema riceve un reward premia le regole utilizzate di recente perché hanno portato a qualcosa di conveniente.
- A certi intervalli di tempo vengono selezionate due regole a cui sono applicati gli operatori genetici.

## Confronto dei Metodi Pittsburgh e Michigan

---

### Pittsburgh

- I geni sono molto lunghi: codificano un insieme di regole.
- La valutazione della fitness richiede molto tempo: valuta un'intera popolazione.
- Richiedono grandi popolazioni e quindi grandi risorse di calcolo.
- È un metodo molto intuitivo.
- Non ci sono interazioni fra regole appartenenti a diverse popolazioni.

### Michigan

- I geni sono molto semplici: codificano una sola regola.
- La fitness viene valutata per ogni regola.
- La popolazione ha la dimensione pari alla base delle regole necessarie.
- Meno intuitivo. Non è chiaro dove sia la parte genetica.
- Le singole regole competono fra di loro in un regime di coabitazione.

## Sistemi a Classificatori

---

Comunemente si intendono con approccio Michigan.

---

1975-1986

Definizione classica di sistema a classificatori.

(LCS Holland 75 & Goldberg 86)

1994-1996

Semplificazione dello schema. (ZCS e XCS Wilson)

## Classificatori (1)

---

- Regole di produzione del tipo IF-THEN indicate con:  
Condizione : Azione
- La condizione è una stringa di lunghezza fissa formata da simboli {0, 1, #}.
- Il simbolo # è detto don't care e indica la possibilità di avere indifferentemente uno 0 o un 1 nella corrispondente posizione.
- L' azione è una stringa binaria di lunghezza fissa.
- Esempio

il classificatore #011 : 01

equivale alla regola

IF Input IS 0011 OR 1011 THEN Action IS 01

## Classificatori (2) - Parametri

---

### Prediction o Strength ( $P_j$ o $S_j$ )

Misura quanto reward riceverò mediamente se utilizzerò quel classificatore.

### Prediction Error ( $\epsilon_j$ )

Misura quanto è corretta la valutazione del parametro di prediction ( $P_j$ ) del classificatore.

### Fitness ( $F_j$ )

Parametro utilizzato per gli operatori genetici.

È alta tanto più precisa la previsione fatta dalla prediction ( $P_j$ ) del classificatore.

## Sistema a Classificatori - Schematizzazione

### Performance Component

- Riassume il funzionamento generale del sistema.

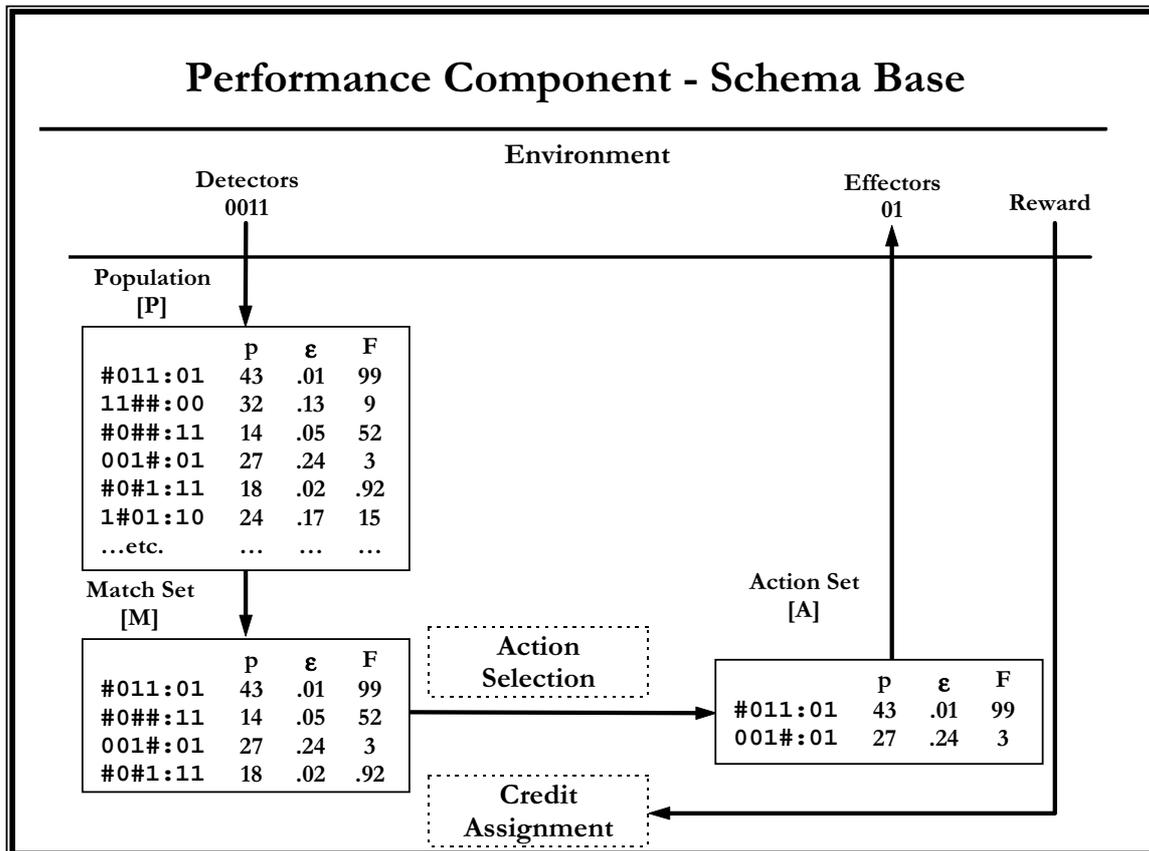
### Reinforcement Component

- Mostra come il reward viene utilizzato per premiare le regole che hanno portato al premio.

### Discovery Component

- Procedure utilizzate per creare nuove regole a partire da quelle esistenti o da condizioni particolari.

## Performance Component - Schema Base



## Costruzione dell'Action Set - Action Selection

Partendo dal Match Set [M] viene costruito il Prediction Array in cui per ogni azione è contenuta una valutazione del reward che il sistema potrebbe ottenere utilizzando quell'azione.

La Prediction di ogni azione  $a$  viene calcolata secondo la formula:

$$P(a) = \frac{\sum_{Action(C_j)=a} p_j F_j}{\sum_{Action(C_j)=a} F_j}$$

Match Set [M]

	P	ε	F
#011:01	43	.01	99
#0##:11	14	.05	52
001#:01	27	.24	3
#0#1:11	18	.02	.92

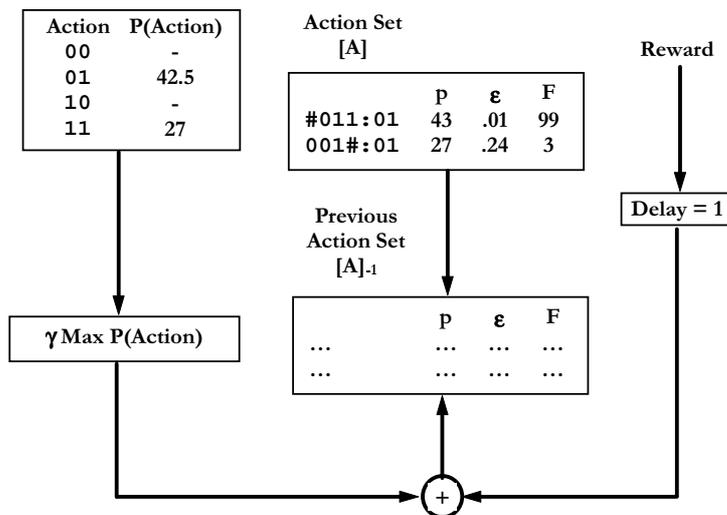
Prediction Array

Action	P(Action)
00	-
01	42.5
10	-
11	27

Roulette Wheel

$$P(01) = \frac{43 \times 99 + 27 \times 3}{99 + 3}$$

## Assegnamento del Reward (1)



## Assegnamento del Reward (2)

---

- Ogni classificatore che ha portato ad un reward viene premiato.
- Il reward viene distribuito sul parametro di prediction del classificatore.
- La formula di aggiornamento utilizzata è identica a quella vista nel Q-learning.

$$p_j = p_j + \beta(\text{reward}_{-1} + \gamma \cdot \max_{a \in \text{Actions}} P(a) - p_j)$$

- $\beta$  corrisponde all' $\alpha$  del Q-learning.
- $\gamma$  ha lo stesso significato che aveva nel Q-learning.
- $p_j$  corrispondono ai Q-value.

## Discovery Component

---

### Covering

Quando non ci sono classificatori la cui parte condizione verifica la configurazione di ingresso.

La condizione è scelta in modo da verificare la configurazione dei sensori

L'azione è scelta in maniera casuale.

### Algoritmo Genetico

A certi intervalli di tempo scelgo due regole e gli applico gli operatori di crossover e mutazione creando due nuove regole.

I parametri delle nuove regole sono posti come media dei parametri dei genitori.

## La Componente Genetica

---

### Algoritmo Genetico

Vengono scelti due classificatori dal Match Set oppure dall'Action Set proporzionalmente alla loro Fitness.

L'algoritmo genetico agisce per nicchie.

Crossover	# 0 #	1 : 1 1	P	$\epsilon$	F
	# 0 1	1 : 0 1	18	.02	92
			43	.01	99
↓					
Nuovi Classificatori	# 0 #	1 : 0 1	P	$\epsilon$	F
	# 0 1	1 : 1 1	30.5	.02	95.5
			30.5	.02	95.5
↓					
Mutazione	# #	1 : 0 1	P	$\epsilon$	F
	# 0 1	1 : 1 1	30.5	.02	95.5
			30.5	.02	95.5

## Explore/Exploit

---

L'Animat, durante ogni esperimento, può:

- Utilizzare ciò che già ha imparato e quindi agendo in Exploitation.
- Continuare a imparare agendo in Exploration.

Provando, ogni tanto, azioni a caso.

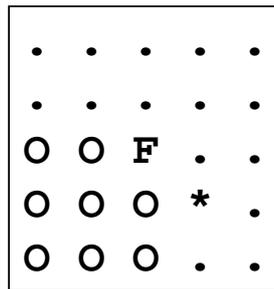
Utilizzando l'algoritmo genetico.

## Esempio - Il problema dell'ANIMAT (1)

ANIMAT o Artificial Animal si muove in un ambiente alla ricerca di cibo.

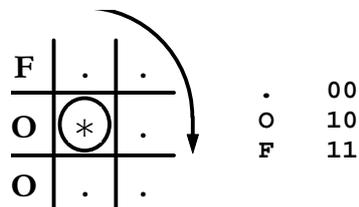
Lo Scopo      Imparare la strada più breve al cibo.

L'Ambiente    WOODS1 è ottenuto ripetendo lo stesso pattern infinitamente in tutte le direzioni.



## Esempio - Il problema dell'ANIMAT (2)

I Sensori      Otto sensori che percepiscono cosa c'è nelle celle adiacenti all'animat.

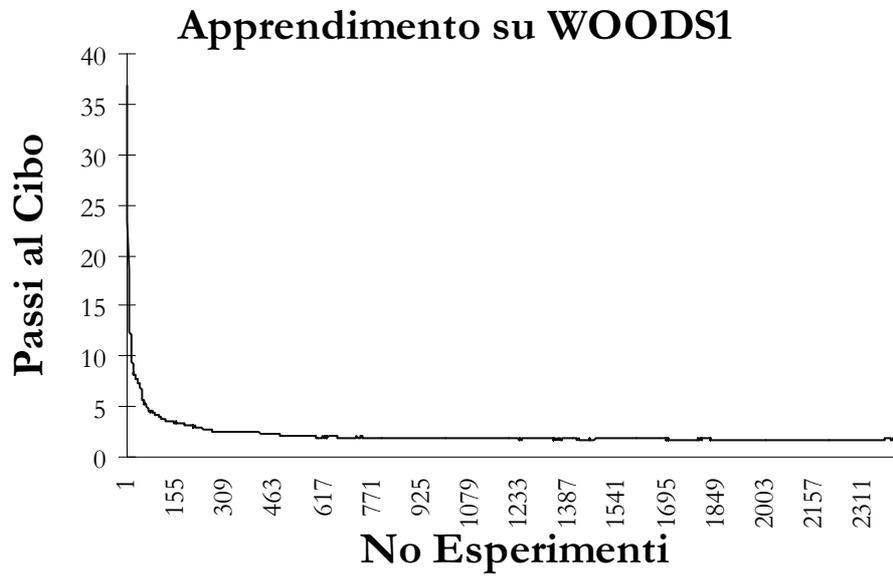


Corrispondente  
Configurazione Sensoriale  
00 00 00 00 00 10 10 11

Il Reward      Ogni volta che l'animat raggiunge il cibo riceve un rinforzo pari a 1000.

### Esempio - Il problema dell'ANIMAT (3)

#### Risultati dell'Apprendimento



### Esempio - Il problema dell'ANIMAT (4)

#### Cosa Impara l'Animat

*	*	*	*	*
*	*	*	*	*
O	O	F	*	*
O	O	O	*	*
O	O	O	*	*

## Esempio - Il problema dell'ANIMAT (4)

○	○	F		
○	○	○		
○	○	○		

## Esercizio - Definizione di un CS (1)

L'ambiente

○	○	○	○	○	○	○	○	○
○	*	.	.	.	.	.	F	○
○	○	○	○	○	○	○	○	○

Codifica delle Condizioni e delle Azioni

Cosa deve percepire l'Animat per imparare il compito assegnato?

Come deve agire?

## **Esercizio - Definizione di un CS (2)**

---

### **Prima Soluzione**

**Si utilizza la stessa rappresentazione  
utilizzata per WOODS1.**

**Otto sensori.**

**Otto azioni.**

---

**Cosa Impara l'Animat?**

## **Esercizio - Definizione di un CS (3)**

---

### **L'Animat Impara**

**A non andare in alto o in basso dato che ci sono i  
muri.**

**A non considerare i sensori relativi alle direzioni  
che non saranno utilizzate.**

---

**L'Animat Impara più cose di quante siano  
necessarie per adempiere al compito  
assegnatogli.**

## **Esercizio - Definizione di un CS (4)**

---

### **Animat Semplificato**

**Due Azioni. Destra e Sinistra.**

**Due sensori uno per il lato Destro e uno per il lato Sinistro.**

**I classificatori hanno 4 bit di parte condizione 2 bit di parte azione.**

## **Sistemi a Classificatori in Ambiente Continuo (1)**

---

### **Agente**

**KHEPERA un robot “da scrivania”.**

### **Sensori**

**Otto sensori di distanza che ritornano un valore compreso fra 0 (lontano) e 1023 (vicino).**

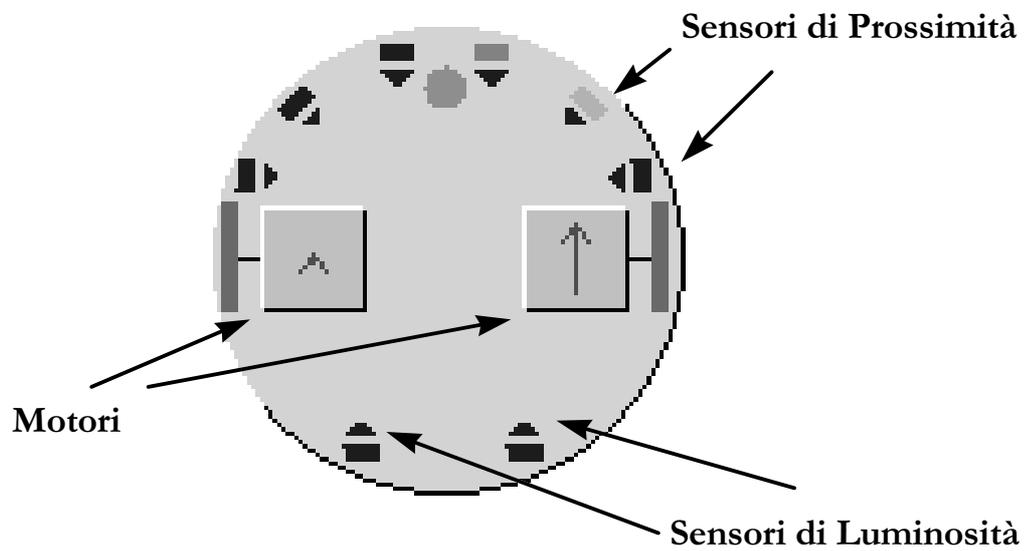
**Otto sensori di luminosità che ritornano un valora compreso fra 0 (buio) e 511 (luminoso)**

### **Attuatori**

**Due motori indipendenti che vengono comandati con due valori compresi fra -5 (indietro) o +5 (avanti)**

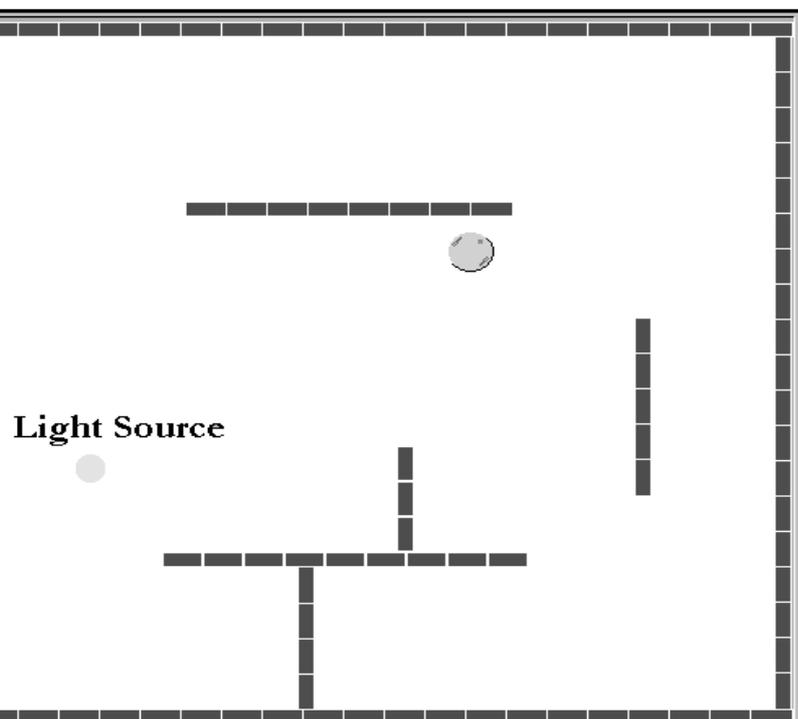
## Sistemi a Classificatori in Ambiente Continuo (2)

### KHEPERA



## Sistemi a Classificatori in Ambiente Continuo (3)

### Ambiente



## **Sistemi a Classificatori in Ambiente Continuo (3)**

---

### **Possibili Compiti**

- **Raggiungere la sorgente di luce.**
  - **Navigare evitando gli ostacoli.**
  - **Seguire una parete**
- 

**A seconda dei diversi compiti utilizzeremo differenti tipi di sensori in differenti posizioni.**

## **Sistemi a Classificatori in Ambiente Continuo (4)**

---

### **Passi di progettazione**

#### **Mappa Condizione/Sensori**

- **Scegliere i sensori che si voglio utilizzare e la relativa codifica.**
- **Scegliere il tipo di azioni che si possono intraprendere.**

#### **Apprendimento**

- **Capire che tipo di apprendimento si vuole utilizzare negli esperimenti.**
- **Quando e come dare il reward.**
- **La strategia di Explore/Exploit da utilizzare.**

## Esempio - Raggiungere una sorgente di luce

---

### Mappa Condizione/Sensori

- Utilizziamo solo i sensori di luminosità.
- Non tutti. Solo I quattro anteriori.
- Codifichiamo solo tre azioni possibili: avanti, un pó a destra (sul posto), un pó a sinistra (sul posto) utilizzando quattro bit.
- 00,01 per avanti; 10 destra, 11 sinistra.

### Apprendimento

- Il problema è simile a quello dell'Animat che cerca cibo.
- Il rinforzo viene dato alla fine quando KHEPERA raggiunge la luce.
- Ogni volta che raggiunge la luce l'agente viene riposizionato casualmente nell'ambiente.

## Esempio - Evitare gli Ostacoli

---

### Mappa Condizione/Sensori

- Utilizziamo solo i sensori di prossimità.
- Tutti e sei quelli anteriori/laterali.
- Non usiamo tutti i bit ma solo una parte (discretizzazione).
- Codifichiamo solo tre azioni possibili come nel caso precedente.

### Apprendimento

- Il problema è completamente diverso dai precedenti in quanto non esiste una vera e propria fine, l'Animat continua ad andare in giro cercando di evitare gli ostacoli.
- Un rinforzo positivo (un premio) può essere dato ogni qualvolta muovendosi l'agente non urta un ostacolo.
- Un rinforzo negativo (una punizione) può essere dato ogni qualvolta l'agente urta un ostacolo.

## Esempio - Seguire un Parete

---

### Mappa Condizione/Sensori

- Utilizziamo solo i sensori di prossimità frontali e quelli laterali relativi alla parete da seguire.
- Codifichiamo solo tre azioni possibili come nel caso precedente.

### Apprendimento

- Non esiste il concetto di esperimento e di fine (magio il cibo, raggiungo la luce). L'agente continua a seguire la parete per un certo periodo.
- Un rinforzo positivo (un premio) può essere dato ogni qualvolta l'agente si trova ad una certa distanza dalla parete.
- Un rinforzo negativo (una punizione) può essere dato ogni qualvolta l'agente si allontana troppo o si avvicina troppo alla parete.

## Estensioni dei Sistemi a Classificatori

---

Le parti condizione/azione sono espresse utilizzando linguaggi più complessi quali:

- S-Espressioni LISP come nel caso della Genetic Programming. Si parla di S-classificatori.
- Condizioni FUZZY. Si parla di classificatori fuzzy.

Si aggiungono componenti interne al sistema come ad esempio strutture di memoria aggiuntive.