# Reinforcement Learning

Dove studieremo alcuni approcci di apprendimento che consentono ad un agente di imparare un comportamento per esperienza, cioè sperimentando

# RL: materiale

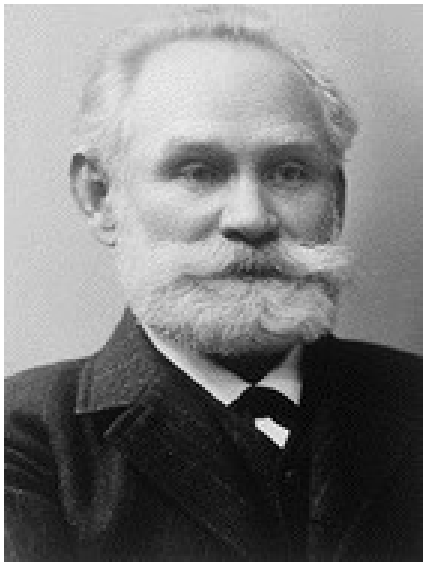
- *Reinforcement Learning: An Introduction*, di Richard S. Sutton e Andrew G. Barto, disponibile on-line, http://incompleteideas.net/book/RLbook2020.pdf: estratti (molto semplificati) dai capitoli 1, 3 (fino a eq 3.9), 6 (solo TD(0))
- Andrew Barto: http://www-all.cs.umass.edu/~barto/
- Rich Sutton: http://www.incompleteideas.net/
- Per chi vuole cimentarsi con qualche sperimentazione: RL-Glue https://glue.rl-community.org/Home

# Condizionamento, ricordate?

Cane affamato

campanello


cibo



Esperimenti sul condizionamento condotti nei primi anni del XX secolo

# Condizionamento

La presentazione del cibo viene ripetutamente accompagnata dal suono di un campanello

Din!!

Cane affamato

campanello

(**stimolo neutro**)


cibo

(**stimolo incondizionato**)

# Condizionamento del comportamento riflesso

Il cane saliva abbondantemente al solo suono del campanello (**stimolo condizionato**)

Stimolo -> Risposta

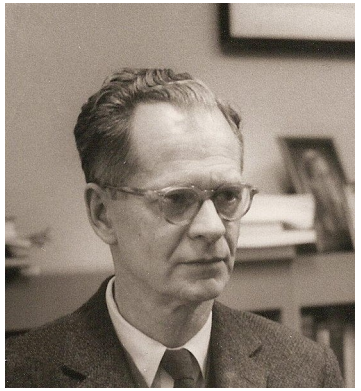
Din!!

L'azione avviene come **effetto condizionato** di uno stimolo inizialmente neutro per il cane.

Condizionamento avvenuto tramite un processo associativo

Il **comportamento riflesso** del cane è stato modificato

# Burrhus Skinner: Condizionamento operante



modalità attraverso la quale l'organismo "apprende"

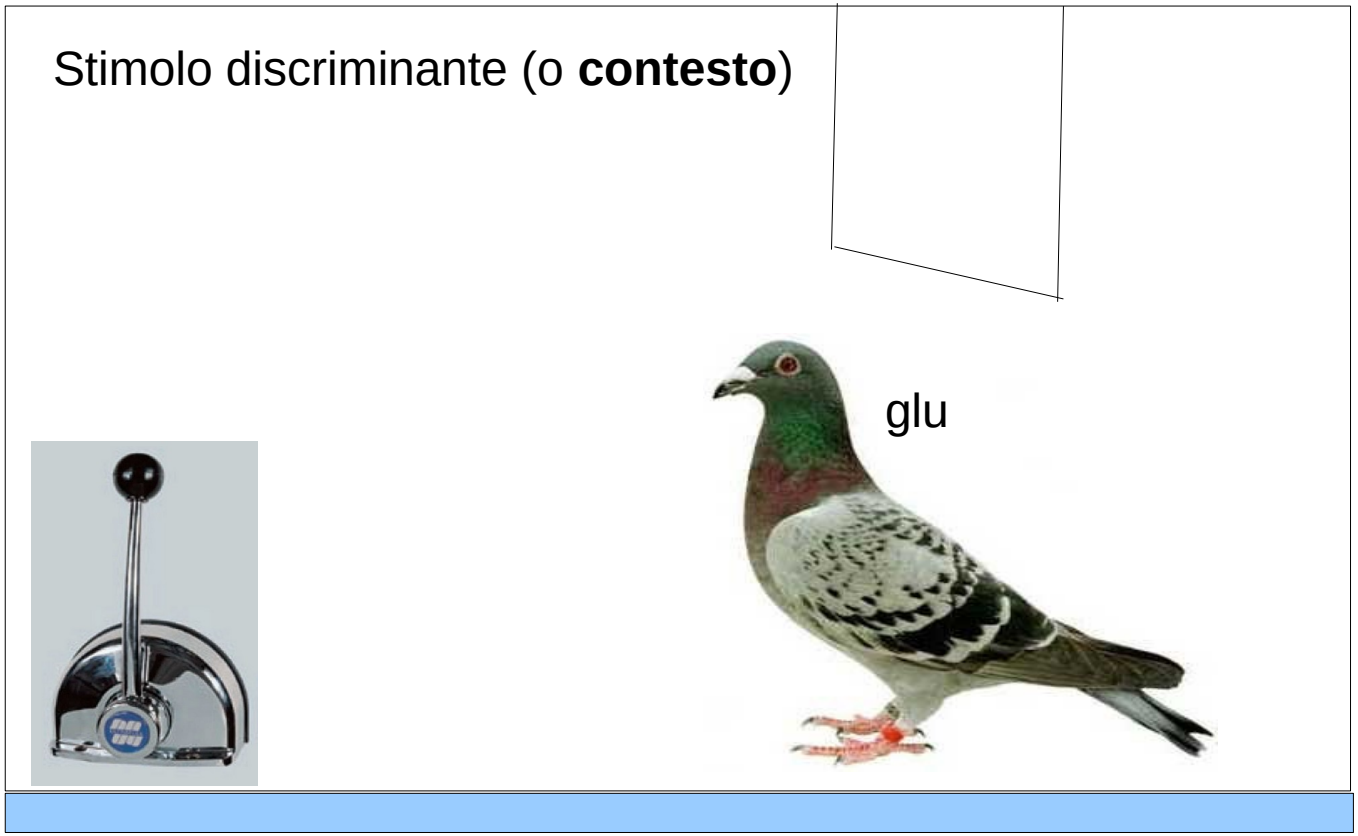
L'apprendimento avviene spesso in modo **non graduale** (*tutto o nulla*)

**Concetti principali**: stimolo discriminante, risposta comportamentale, **stimolo rinforzatore**

**Scatola di Skinner**

# Condizionamento operante

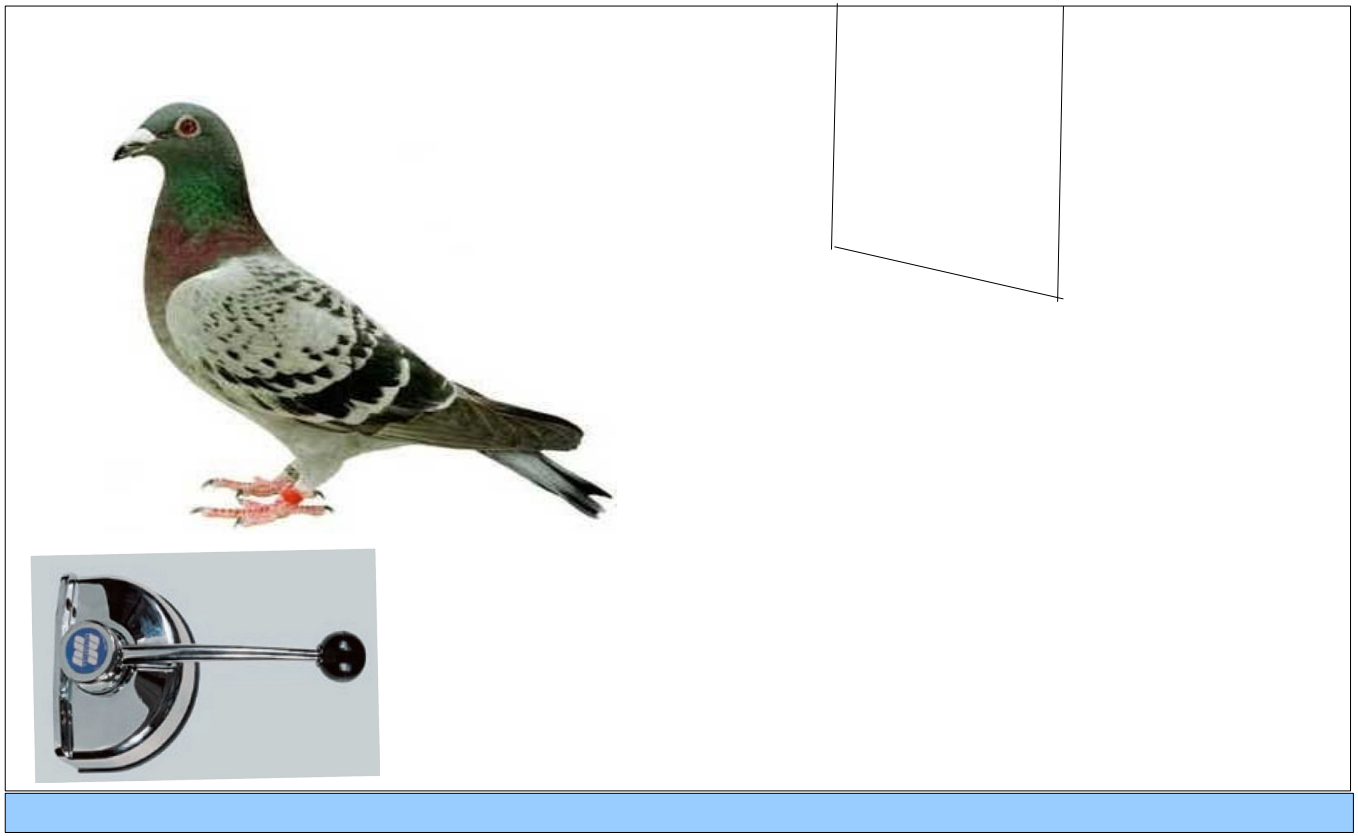
Un piccione viene inserito in un contesto e lasciato libero di fare quello che vuole ...




Scatola di Skinner

# Condizionamento operante

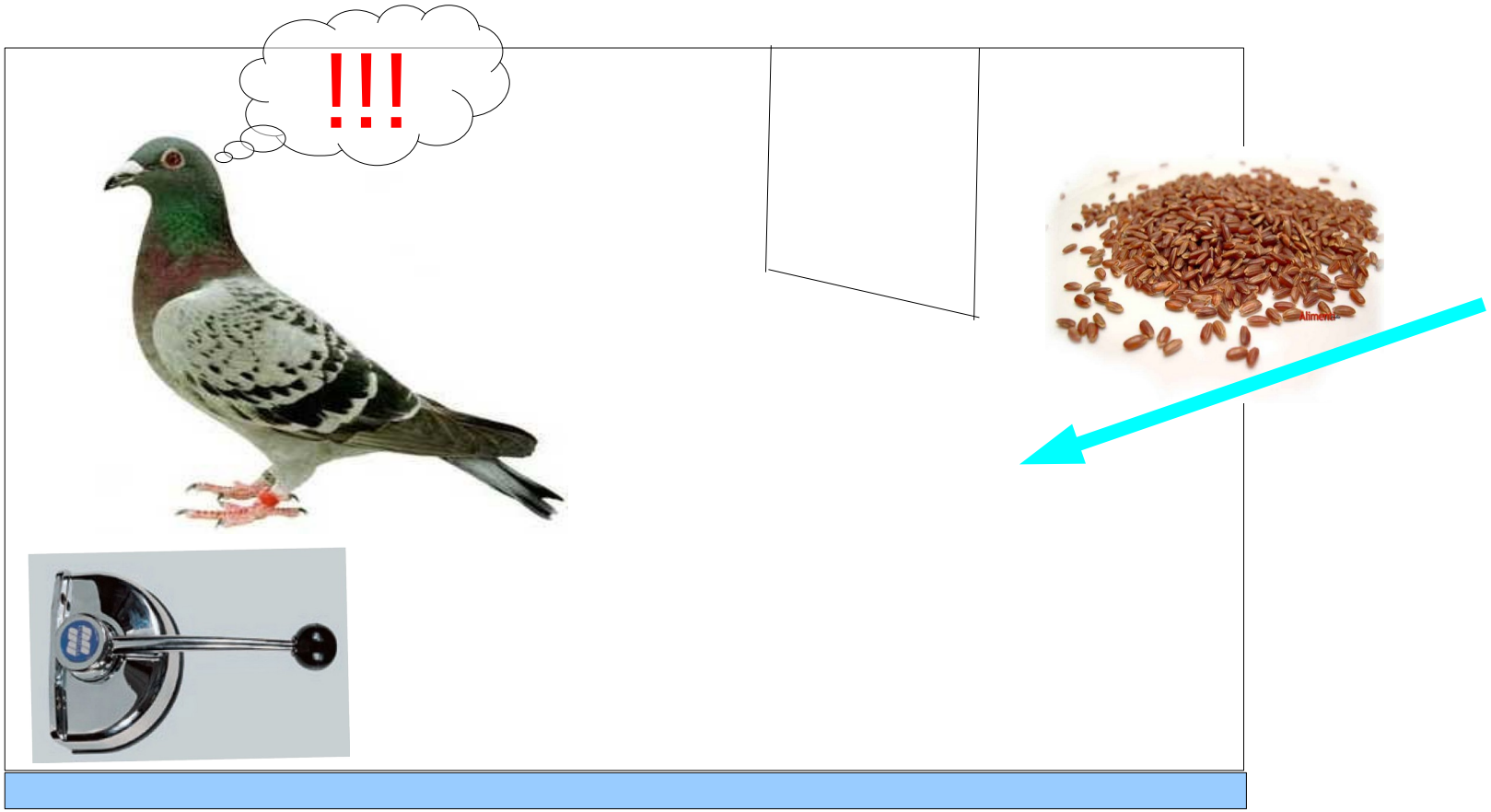
... quando casualmente abbassa una certa leva (**comportamento**), gli viene dato del cibo



Scatola di Skinner

# Condizionamento operante

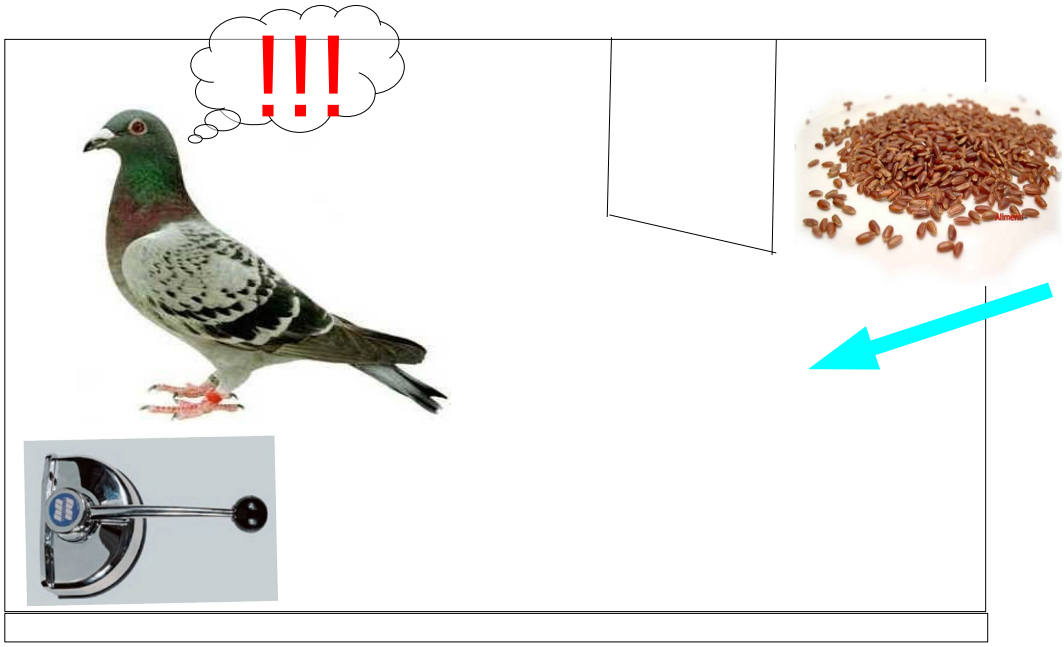
... quando casualmente abbassa una certa leva, gli viene dato del cibo (**stimolo rinforzatore**)




Scatola di Skinner

# Condizionamento operante

Cibo = **premio**




Risposta -> Stimolo

Ricevendo un premio a seguito di determinati comportamenti, il piccione *aumenta la frequenza di quei comportamenti*

Il **condizionamento** è detto **operante** perché il piccione **sceglie di *comportarsi diversamente***. Non si tratta di forzare reazioni non controllabili dall'animale

Il condizionamento operante è un *meccanismo di adattamento all'ambiente* insito negli organismi viventi

**Estinzione operante**: meccanismo analogo che inibisce certi comportamenti tramite la somministrazione di **punizioni** (*rinforzi negativi*)

# Riassumendo

- **Condizionamento**: modifica il comportamento riflesso, non controllato in maniera conscia
- **Condizionamento operante**: modifica il comportamento conscio grazie a un meccanismo di premi/punizioni

- **COLLEGAMENTO**: gli agenti software agiscono in un ambiente, posso causare la costruzione automatica di un “comportamento” introducendo una forma di condizionamento operante anche nei sistemi software?

# Cos'è un comportamento?

**Comportamento**: effetto dell'agire, comporta la scelta delle azioni da eseguire situazione per situazione

**Ambiente**: ciò che non è sotto il controllo dell'agente

**Adattamento all'ambiente**: agente non più unica componente attiva del binomio agente-ambiente; immagino l'**ambiente** come produttore e restitutore di **feedback**

# Reinforcement learning

**Reinforcement Learning** (apprendimento per rinforzo): approccio computazionale all'apprendimento per interazione

Riguarda l'imparare che cosa fare, *associando azioni a situazioni in modo tale da massimizzare un segnale numerico di compenso*

Il discente scopre, in maniera indiretta, come si desidera che si comporti operando per tentativi (**procedimento trial-and-error**).

Cosa guida l'apprendimento? **Massimizzazione del compenso atteso** (qualche volta immediato ma il più delle volte futuro, **ritardato** rispetto al momento in cui l'agente decide come comportarsi)

# Reinforcement learning

In queste lezioni studieremo un approccio computazionale all'apprendimento per interazione detto **Reinforcement Learning** (apprendimento per rinforzo).

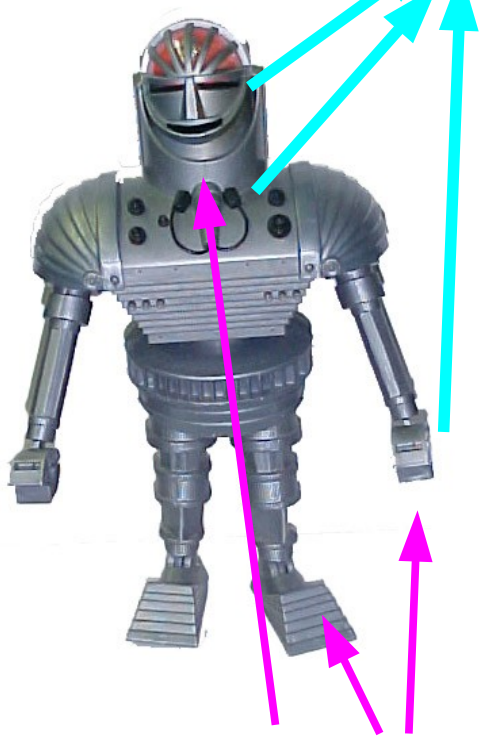
Il reinforcement learning riguarda l'imparare che
*azioni a situazioni in modo tale*

Il disc
(**proce**

Il suo a
(qualch
cui l'agente decide come comportarsi).

Se abbiamo un comportamento desiderato che vogliamo che l'agente assimili, perché non scriviamo un programma che codifica questo comportamento? O perché non usiamo tecniche di planning?

# Controllori



**Sensori**: permettono di percepire l'ambiente

**Controllore**: programma che sceglie l'azione da compiere

**Attuatori**: permettono di agire

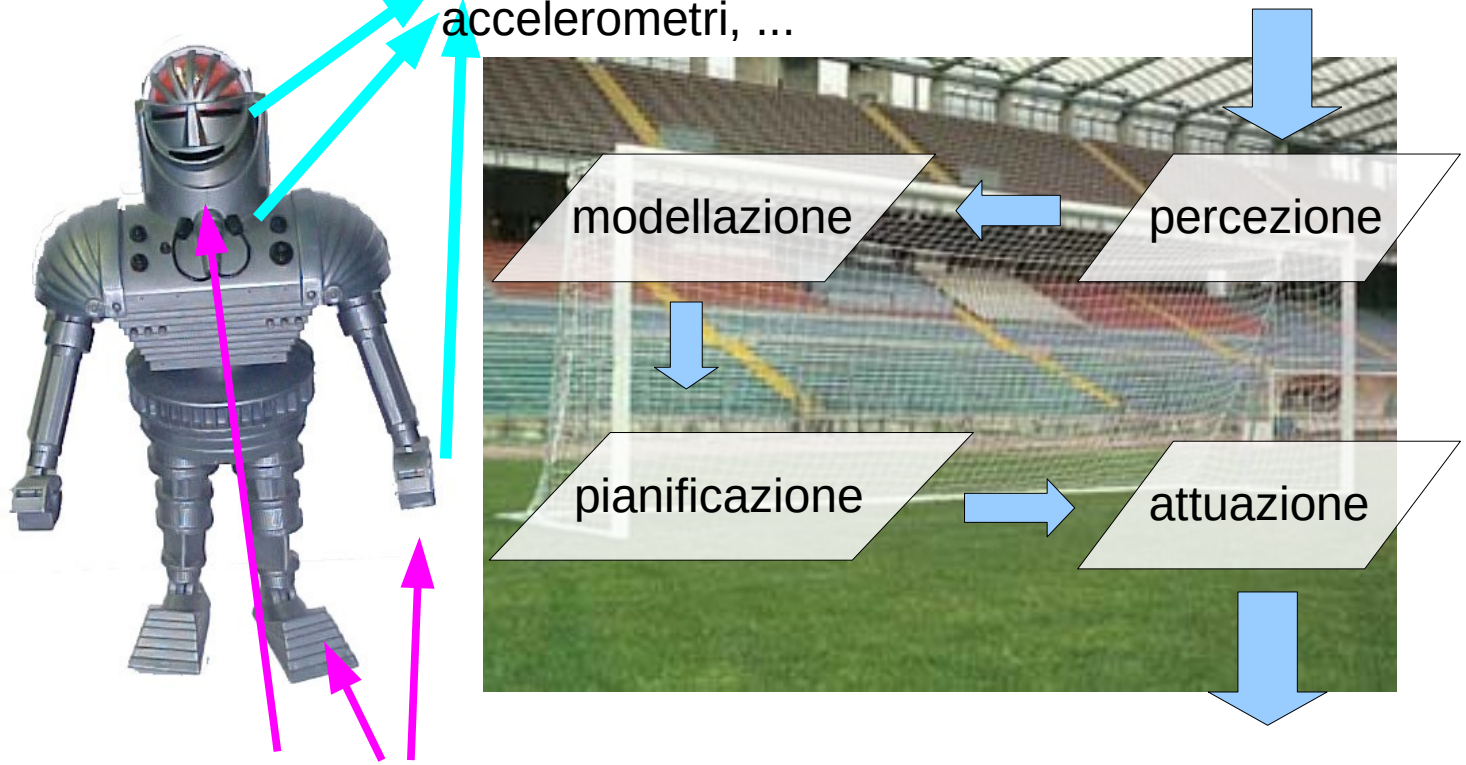
# Scrivo controllore in C per parare un rigore

**Sensori**: telecamere, sensori di distanza, di contatto, accelerometri, ...


**Attuatori**: giunti articolari con più gradi di libertà

# Uso un planner per parare un rigore

**Sensori**: telecamere, sensori di distanza, di contatto, accelerometri, ...

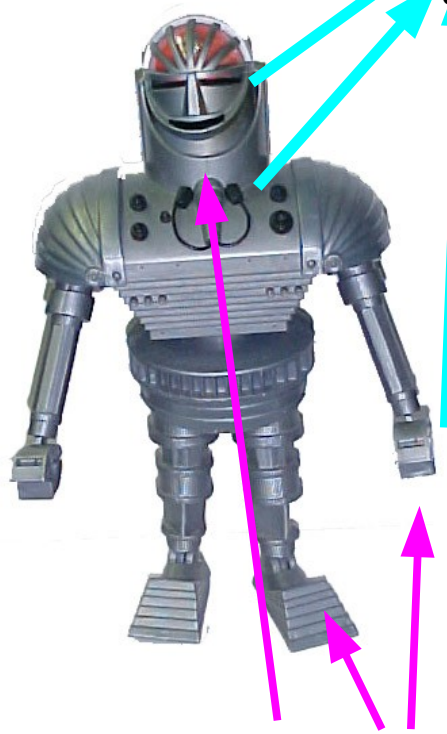



**Attuatori**: giunti articolari con più gradi di libertà

**Pianificazione**: attività razionale, spesso non adatta a problemi da affrontare in real-time (time-consuming)

# Apprendimento per rinforzo

**Sensori**: telecamere, sensori di distanza, di contatto, accelerometri, ...



**Processo trial-and-error**

**Goal: rinforzo -1**

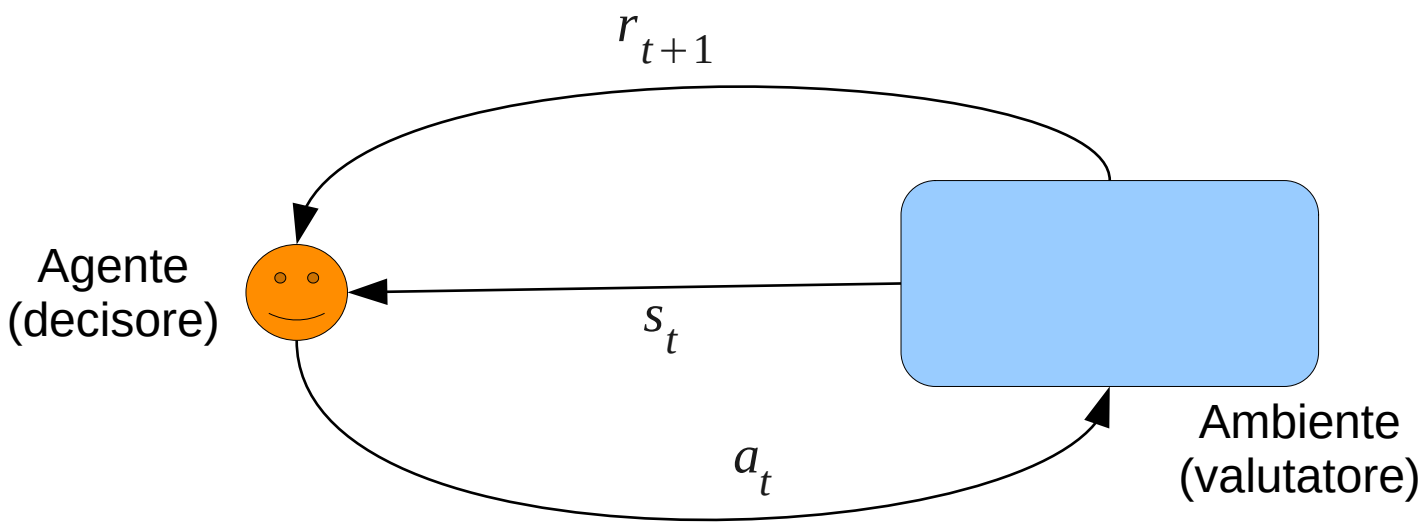
**Parata: rinforzo +1**

L'agente costruisce il controllore in autonomia

**Attuatori**: giunti articolari con più gradi di libertà

**A carico del programmatore**: scrivere la funzione di rinforzo, dopo aver identificato gli stati di successo e quelli di fallimento (e tanto tempo per l'addestramento)

# Agente-Ambiente




L'interazione avviene ad ogni istante di una sequenza temporale *t = 1, 2, 3, ...*

Ad ogni istante l'agente:

1) riceve una rappresentazione della situazione corrente ($s_t$)
2) Sceglie un'azione $a_t$ fra quelle eseguibili e la esegue

All'istante successivo l'agente riceve un rinforzo numerico ($r_{t+1}$) e tutto ricomincia

Vedere: http://incompleteideas.net/sutton/book/ebook/node28.html

# Policy

**Mapping**:

$s_1$ -> $p\_a_1$ $p\_a_2$ ... $p\_a_n$
$s_2$ -> $p\_a_1$ $p\_a_2$ ... $p\_a_n$
...
$s_n$ -> $p\_a_1$ $p\_a_2$ ... $p\_a_n$

L'agente implementa un mapping fra situazioni ($s_j$) e azioni possibili ($a_i$): il valore $p\_a_i$ è la probabilità di selezionare l'azione $a_i$ essendo nella situazione $s_j$

Il mapping è detto **policy** ed è denotato da $\pi_t$

$\pi_t$ (s, a): probabilità di selezionare l'azione *a* essendo nella situazione *s*

**Scopo dell'apprendimento**: costruire una policy che consenta all'agente di massimizzare il compenso totale ricevuto durante il suo funzionamento (obiettivo dell'agente).

# Note

- **Framework astratto e flessibile**:
  - gli istanti temporali non necessariamente identificano intervalli fissi di tempo fisico;
  - le azioni possono essere di basso livello (es. accelerazioni o voltaggi) o astratte (es. andare a pranzo);
  - gli stati possono essere puramente percettivi (output dei sensori) o simbolici (descrizioni di oggetti, di luoghi);
  - gli stati possono essere soggettivi, collettivi, comprendere una memoria del passato;
  - le azioni possono essere fisiche o “mentali” (es. Scegliere a cosa prestare attenzione)

- **Confini fra agente e ambiente:**

  spesso le parti meccaniche di un agente robotico sono considerate parti dell'ambiente;

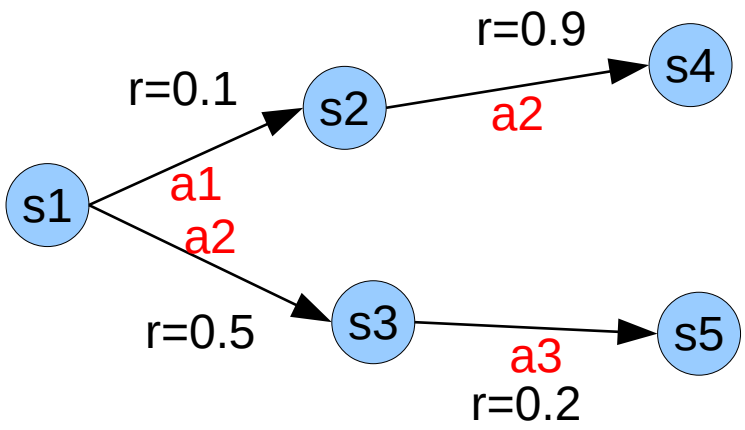
- **Compensi/punizioni:**

  anche se negli esseri biologici sono prodotti dall'agente stesso, in RL sono considerati come un prodotto dell'ambiente;

- *In generale tutto ciò che non può essere modificato arbitrariamente dall'agente è considerato parte del suo ambiente*

# Goal e compensi

Il **goal** dell'agente è massimizzare il compenso **ricevuto complessivamente** (NB: non quello immediato, ricevuto stato per stato)




Se r è il compenso immediato per ognuna delle azioni, scegliere a1 quando si è in s1 è sconveniente dal punto di vista del guadagno immediato ma è conveniente in termini di guadagno complessivo

Il compenso indica ***cosa*** si desidera che l'agente faccia non il ***come*** *farlo*

# Rinforzi immediati, esempi

- Per far sì che un robot impari a camminare si usano, ad ogni passo, rinforzi proporzionali alla quantità di movimento in avanti che è riuscito a fare;
- Per far sì che un agente impari a uscire da un labirinto spesso il compenso è 0 (nessun compenso) finché rimane prigioniero, +1 quando trova l'uscita;
- In alternativa spesso lo si punisce (compenso -1) per ogni istante che trascorre nel labirinto. In questo modo l'agente è sollecitato a trovare l'uscita il più in fretta possibile;
- Per insegnare a un robot a raccogliere lattine, gli si fornisce un compenso che è sempre zero tranne quando trova una lattina buttata. In questo caso il compenso è +1;
- Talvolta gli si dà anche un compenso negativo (-1) ogni volta che sbatte contro qualcosa o che raccoglie un oggetto sbagliato;
- Se un agente deve imparare a giocare a scacchi o a dama gli si può dare +1 in caso di vittoria, -1 in caso di sconfitta e 0 per tutte le posizioni non terminali.

# Compenso atteso

$$R_t = r_{t+1} + r_{t+2} + \ldots + r_{t+n}$$

$t$ = istante corrente

$r_{t+j}$ = compenso che si riceverà all'istante t+j

$t+n$ = istante terminale dell'esecuzione

***n*** **< ∞** allora l'interazione con l'ambiente ha **natura episodica**
è possibile separare gli stati terminali da quelli non terminali

***n*** **== ∞** allora l'esecuzione è continua, la somma tende essa stessa a diventare infinita. Ciò porta all'introduzione del concetto di ***discount*** **$0 \leq \gamma \leq 1$**:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^{n-1} r_{t+n} = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}$$

Un rinforzo ricevuto fra *k* istanti, in questo momento ha un valore scontato di $\gamma^{k-1}$

http://incompleteideas.net/sutton/book/ebook/node30.html

# Note

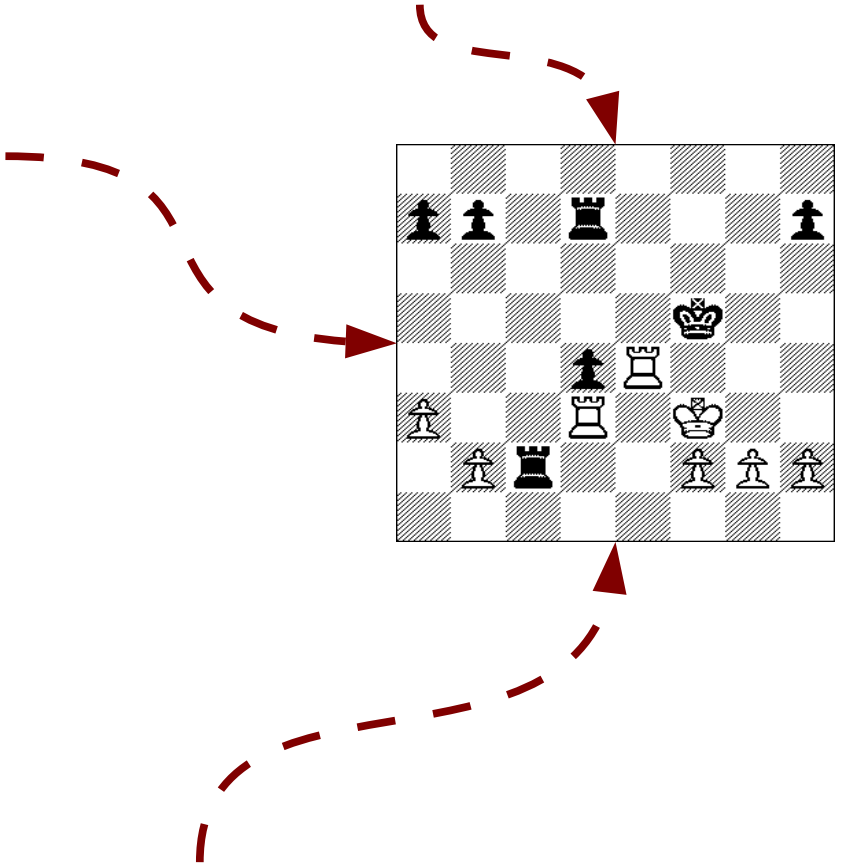
- Se $\gamma = 0$ allora l'agente è miope: $R_t = r_{t+1}$

  non è in grado di considerare i rinforzi futuri, usa solo il rinforzo immediato (rincorre il guadagno immediato);

- Via via che $\gamma$ cresce e si approssima ad 1 l'agente tiene sempre più in considerazione il futuro

# Indipendenza dal cammino

- In generale un agente può essere visto come una funzione **a = fz(s)** che, dato uno stato *s* produce un'azione *a*
- L'agente non ha altra informazione che *s* per fare la sua scelta



Nei problemi che affrontiamo *non è importante* ***come*** si è pervenuti allo stato corrente per effettuare la scelta dell'azione

Lo stato riassume tutta l'informazione vista rilevante

Si dice che lo stato è **Markoviano** o che è *indipendente dal cammino*

# Markov decision process

Un compito di RL che soddisfa la proprietà Markoviana è detto **processo di decisione Markoviano** (MDP). Se l'insieme dei suoi stati e quello delle sue azioni sono finiti, si ha un **MDP finito**. Gli MDP finiti sono molto importanti nella teoria del RL.

Un **MDP finito** è definito da:

**S**: insieme degli stati
**A**: insieme delle azioni
$\mathbf{P}^{a}_{ss'}$: probabilità di transizione (probabilità di transire dallo stato *s* a *s'* eseguendo *a*)
$\mathbf{R}^{a}_{ss'}$: valore atteso del prossimo reward (valore di compenso atteso eseguendo a essendo nello stato s e raggiungendo lo stato s')

$$P^{a}_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \qquad R^{a}_{ss'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$$

**Perché probabilità e non certezze?**
Perché non è detto che la nostra conoscenza degli stati sia completa

# Esempio di MDP finito: robot spazzino

**LEGGERE (VEDERE ANCHE LIBRO)**

Dustcart
https://en.wikipedia.org/wiki/Dustbot

L'agente ha lo scopo di raccogliere lattine vuote lasciate per terra. In ogni istante deve decidere se cercare una lattina, aspettare che qualcuno faccia qualcosa, andare a ricaricare la batteria. Quando la battria si scarica l'agente non può che aspettare di essere recuperato.

Supponiamo che l'agente debba prendere decisioni basandosi solo sulla carica della batteria (**high** oppure **low**).

Le azioni possibili sono: **wait**, **search**, **recharge**.

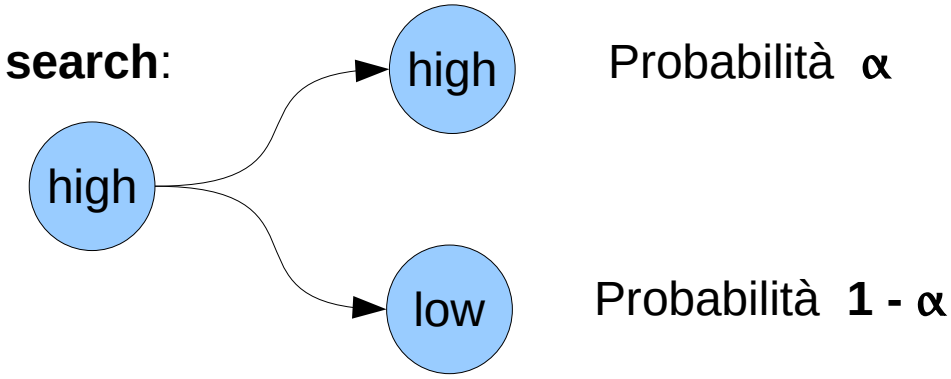
Se la carica è **low**, sono possibili le azioni **wait**, **search** e **recharge**

Se la carica è **high**, recharge non ha senso

(Esempio originale sviluppato da Richard Sutton)

# Esempio (continua)

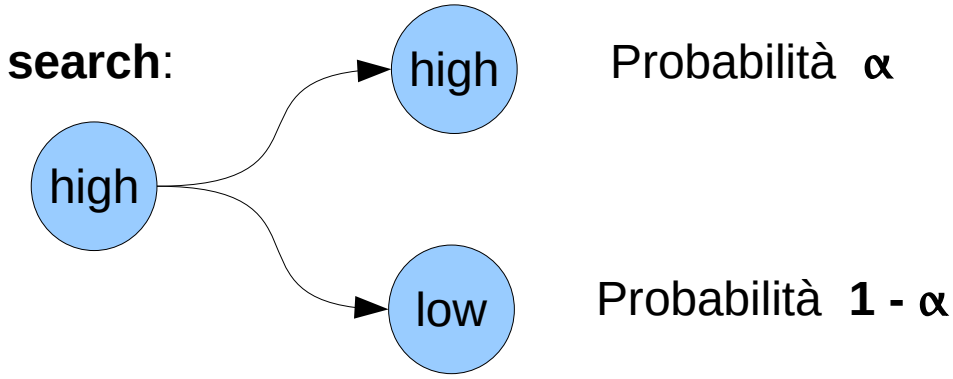
La batteria si consuma in modo continuo, l'agente distingue però solo **high** e **low**




**LEGGERE DA SOLI**

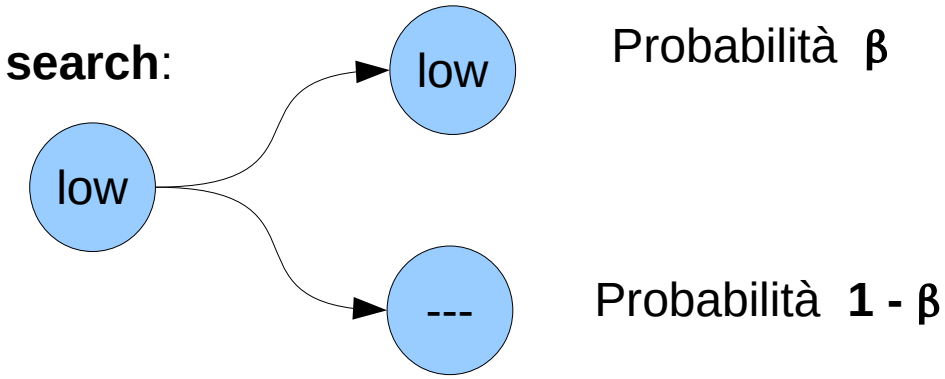
# Esempio (continua)

La batteria si consuma in modo continuo, l'agente distingue però solo **high** e **low**




Probabilità $\alpha$

Probabilità $1 - \alpha$




Probabilità $\beta$

Probabilità $1 - \beta$

Morto: va recuperato, quindi viene ricaricato

**LEGGERE DA SOLI**

# Esempio (continua)

Rinforzi e altre caratteristiche:

**Raccolta di una lattina:** rinforzo +1

**Scaricamento totale batteria:** rinforzo -3

**N_search**: numero lattine raccolte eseguendo search

**N_wait**: numero lattine raccolte eseguendo wait

Non si raccolgono lattine nel tragitto verso la ricarica

**LEGGERE DA SOLI**

# Esempio: probabilità di transizione e compenso atteso

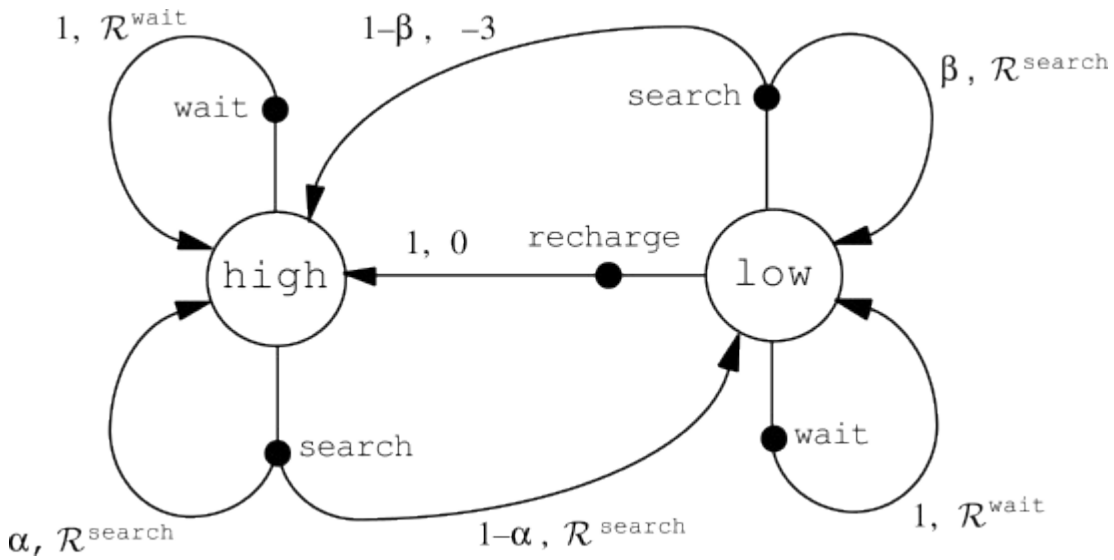
| $s=s_t$ | $s'=s_{t+1}$ | $a=a_t$ | $P^a_{ss'}$ | $R^a_{ss'}$ |
|---|---|---|---|---|
| high | high | search | **α** | N_search |
| high | low | search | **1-α** | N_search |
| low | high | search | **1 - β** | -3 |
| low | low | search | **β** | N_search |
| high | high | wait | 1 | N_wait |
| high | low | wait | 0 | N_wait |
| low | high | wait | 0 | N_wait |
| low | low | wait | 1 | N_wait |
| low | high | recharge | 1 | 0 |
| low | low | recharge | 0 | 0 |

**LEGGERE DA SOLI**

# Esempio di MDP finito

**LEGGERE DA SOLI**

Spesso si usa una rappresentazione a grafo:

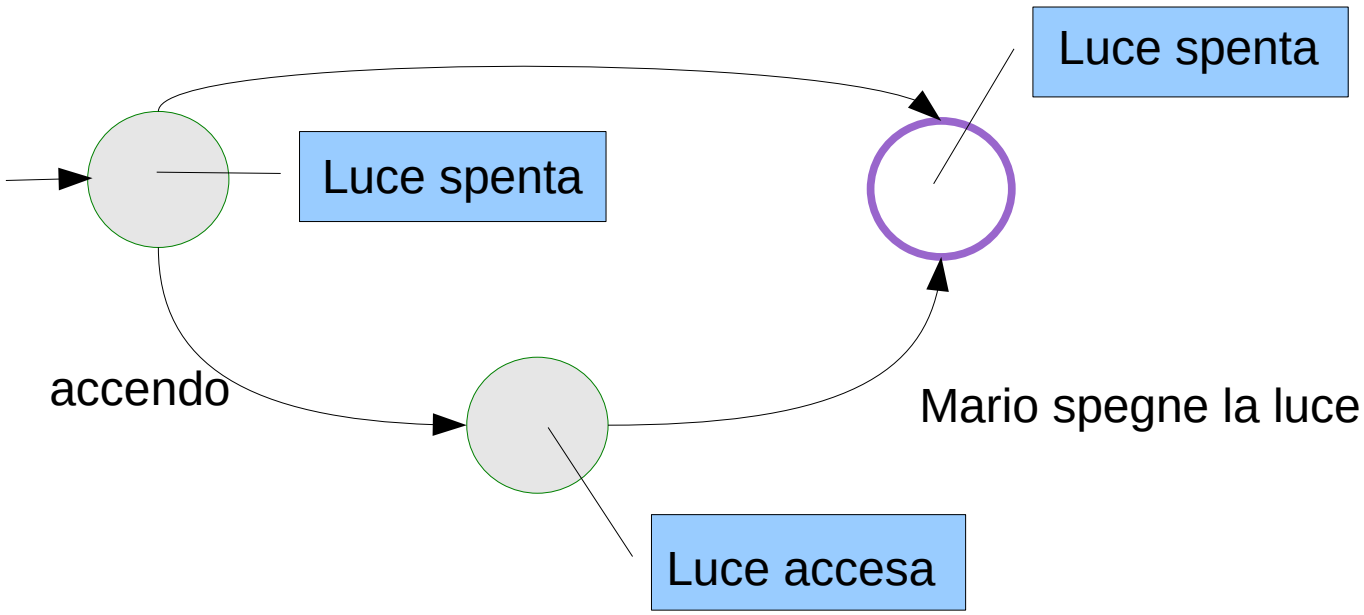



$R^{search}$ = N_search
$R^{wait}$ = N_wait

http://incompleteideas.net/sutton/book/ebook/node33.html

# Non tutti i problemi sono markoviani

**Esempio**: supponiamo che mi sia impegnata (*commitment*) ad accendere la luce e che in questo istante la luce sia spenta. Il problema da risolvere è: ho assolto al mio impegno?




Occorre conoscere la storia per rispondere