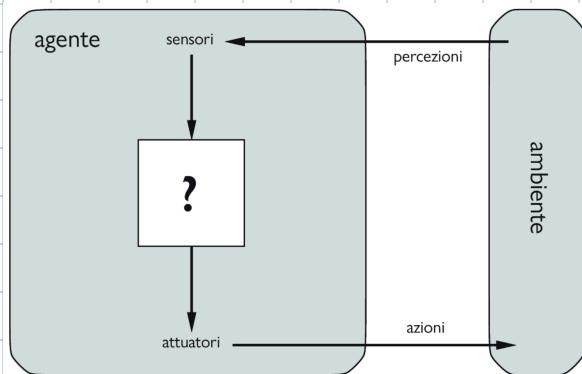


# DEFINIZIONE DI AGENTE

AGENTE → QUAISIASI COSA CHE COSTITUISCE UN SISTEMA CHE PERCEPISCE IL SUO AMBIENTE ATTRAVERSO DEI SENSORI E AGISCE MEDIANTE ATTUATORI.



PERCEZIONI → DATI PERCEPITI DAI SENSORI DI UN AGENTE.

SEQUENZA PERCETTIVA → STORIA COMPLETA DI TUTTO CIÒ CHE ESSO HA PERCEPITO NELLA SUA ESISTENZA.

AZIONE → INTAIPESA DALL'AGENTE IN BASE ALLA SUA CONOSCENZA OPPURE ALLA SUA SEQUENZA PERCETTIVA, ANDIAMO QUINDI A SPECIFICARE L'AZIONE PER OGNI POSSIBILE SEQUENZA PERCETTIVA.



FUNZIONE AGENTE → DESCRIVE IL COMPORTAMENTO DI UN AGENTE DESCRIVENDO LA CORRISPONDENZA TRA UNA SEQUENZA PERCETTIVA E UNA SPECIFICA AZIONE, È UNA DESCRIZIONE ASTRATTA.



PUÒ ESSERE RAPPRESENTATA CON:

PROGRAMMA AGENTE → RAPPRESENTA IN FORMA TABELLARE LA FUNZIONE AGENTE, PUÒ ESSERE INFINTA A MENO CHE NON SI FISSI UNA LUNGHEZZA MASSIMA PER LE SEQUENZE PERCETTIVE.

## ESEMPIO

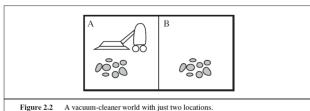


Figure 2.2 A vacuum-cleaner world with just two locations.

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
:	
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck
:	

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

→ TABELLA CHE DESCRIVE LA FUNZIONE AGENTE (PROGRAMMA AGENTE)

[A, Clean], [A, Clean], [A, Clean]  
[A, Clean], [A, Clean], [A, Dirty]

→ SEQUENZA PERCETTIVA

# Agente razionale

Un agente razionale è un'entità che interagisce con il suo ambiente in maniera efficace, ossia prende decisioni in base alle percezioni ricevute dall'ambiente per raggiungere obiettivi specifici.

- **Fa la cosa giusta:** L'agente razionale seleziona le azioni che massimizzano la probabilità di successo o di raggiungere il proprio obiettivo.
- **La sequenza di stati ha un qualche aspetto auspicabile:** L'agente non si limita a scegliere azioni immediate, ma considera anche le conseguenze a lungo termine delle sue azioni, ottimizzando così una sequenza di stati che porta a un risultato favorevole.
- **Mostra una preferenza selettiva verso certi comportamenti:** L'agente razionale non prende decisioni casuali, ma seleziona le azioni che considera migliori rispetto a un determinato criterio o metrica di valutazione (ad esempio, massimizzazione della ricompensa o minimizzazione dei costi).

👤 Per ogni sequenza di percezioni compie l'azione che **massimizza** il valore atteso della misura delle prestazioni, considerando le sue percezioni passate e la sua **conoscenza pregressa**.

## Razionalità

❓ Che significa essere razionale?

È **razionale** quello che ci conviene per un **equilibrio**

- Si considera razionale quello che la maggior parte degli umani fanno (**la misura di intelligenza in natura non esiste**)
- Significa quindi funzionare in modo naturale come unita umane

❓ Problema P: Agente razionale prende il problema e genera una soluzione. Se invece la soluzione è fatta di diversi passi, la soluzione è in base allo svolgimento dei passi ottimo per arrivare a quella soluzione

**PEAS → prestazione, ambiente, attuatori, sensori**

## MISURE DI PRESTAZIONE

AGENTE VIENE INSERITO IN UN AMBIENTE → GENERA SEQUENZA DI AZIONI IN BASE ALLE PERCEZIONI CHE RICEVE.



LA SEQUENZA DI AZIONI PONTE L'AMBIENTE AD ATTRAVERSARE UNA SEQUENZA DI STATI



CATTURABILE CON UNA MISURA DI PRESTAZIONE

## RAZIONALITÀ

AGENTE RAZIONALE SI BASA SU:

- MISURA DI PRESTAZIONE CHE DEFINISCE IL CRITERIO DI SUCCESSO
- CONOSCENZA PREGESSA DELL'AMBIENTE DA PARTE DELL'AGENTE
- LE AZIONI CHE L'AGENTE PUÒ EFFETTUARE
- LA SEQUENZA PERCETTIVA DELL'AGENTE FINO ALL'ISTANTE CORRENTE

## SPECIFICARE UN AMBIENTE

PEAS → PERFORMANCE, ENVIRONMENT, ACTUATORS, SENSORS.

↳ QUANDO SI PROGETTA UN AGENTE IL PRIMO PASSO È SPECIFICARE L'AMBIENTE OPERATIVO (PEAS).

# Ambienti

Gli ambienti rappresentano lo spazio o il contesto in cui un agente interagisce e prende decisioni. L'agente, per funzionare correttamente, deve essere in grado di percepire l'ambiente circostante tramite sensori e agire su di esso tramite attuatori per raggiungere un obiettivo specifico.

## Osservabilità

- **Completamente osservabile:** L'agente può osservare interamente lo stato dell'ambiente in ogni momento. Ha accesso a tutte le informazioni necessarie per prendere decisioni.  
Esempio: un gioco da tavolo come gli scacchi, dove l'agente può vedere tutte le pedine sulla scacchiera.
- **Parzialmente osservabile:** L'agente ha solo una visione parziale dello stato dell'ambiente. Deve prendere decisioni basate su informazioni incomplete o incerte. Esempio: un robot che si muove in una stanza buia con sensori limitati.

## Deterministico vs Stocastico

- **Deterministico:** Le azioni dell'agente determinano esattamente l'esito. Non c'è casualità o incertezza nelle conseguenze delle azioni. Se l'agente esegue un'azione, il risultato sarà sempre lo stesso. Esempio: un sistema di controllo di un ascensore, dove ogni azione produce sempre lo stesso risultato.
- **Stocastico:** Le azioni dell'agente non garantiscono un risultato certo, c'è incertezza. Esempio: un robot che naviga in un ambiente reale potrebbe non avere il pieno controllo sulle sue azioni a causa di ostacoli imprevisti o variazioni nell'ambiente.

## Statico vs Dinamico

- **Statico:** L'ambiente non cambia mentre l'agente prende decisioni. L'agente può concentrarsi solo sull'esecuzione delle sue azioni senza preoccuparsi che l'ambiente cambi nel frattempo.  
Esempio: un problema di risoluzione di un puzzle come il cubo di Rubik.
- **Dinamico:** L'ambiente può cambiare mentre l'agente prende decisioni. L'agente deve essere capace di reagire rapidamente e adattarsi ai cambiamenti. Esempio: un'auto a guida autonoma che si muove in città, dove il traffico e i pedoni cambiano continuamente.

## Discreto vs Continuo

- **Discreto:** L'ambiente ha un numero finito di stati e azioni. Le decisioni dell'agente riguardano scelte in una serie di possibilità ben definite. Esempio: un gioco come il tris.
- **Continuo:** L'ambiente ha un numero infinito o non definito di stati e azioni. Esempio: il controllo di un braccio robotico in uno spazio tridimensionale, dove le posizioni e i movimenti possibili sono infiniti.

## Monoagente vs Multiagente

- **Monoagente:** L'ambiente contiene solo un agente, quindi non c'è competizione o collaborazione con altri agenti. Esempio: un aspirapolvere robot che pulisce una stanza.
- **Multiagente:** L'ambiente contiene più agenti che possono cooperare o competere tra loro. Esempio: una partita di calcio robotico, dove più agenti (robot) lavorano insieme o contro di loro per raggiungere un obiettivo.

## Episodico vs Sequenziale

- **Episodico:** Ogni decisione o azione presa dall'agente è indipendente dalle azioni precedenti. Esempio: un sistema che analizza immagini indipendenti.
- **Sequenziale:** Le azioni dell'agente influenzano le azioni future e la sequenza degli stati. Esempio: la guida di un robot che deve pianificare le sue mosse successive in base alla posizione precedente.

# Programma Agente

```
function SKELETON-AGENT (percept) returns action
  static: memory %the agent's memory of the world
    memory □ UPDATEMEMORY( memory, percept)
    action □ CHOOSE-BEST-ACTION( memory)
    memory □ UPDATEMEMORY( memory, action)
  return action
```

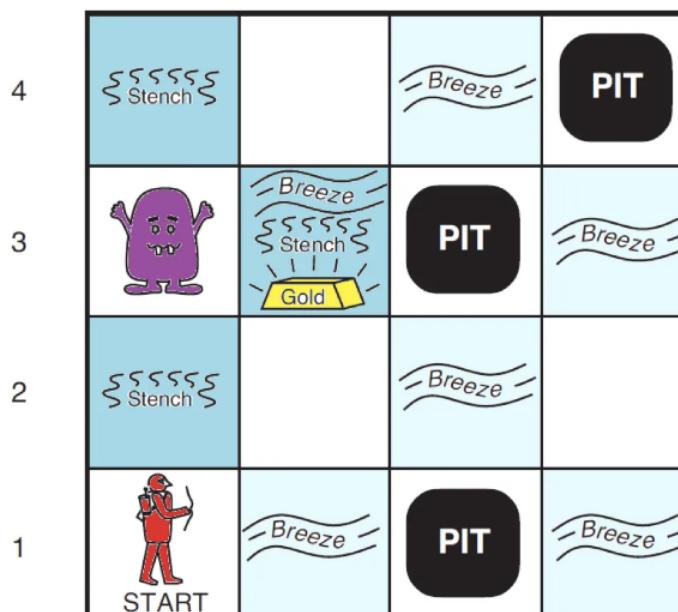
- Il programma dell'agente viene descritto come una funzione scheletrica, chiamata **SKELETON-AGENT**, che prende in ingresso le percezioni (percept) e restituisce un'azione (action).
  - **static: memory** : L'agente mantiene una **memoria statica**. Questa memoria è utilizzata per conservare informazioni sullo stato dell'ambiente, delle azioni passate o altri dati utili per prendere decisioni future.
  - La prima operazione è **aggiornare** la memoria dell'agente in base alle percezioni ricevute. Questo avviene con la funzione **UPDATE-MEMORY(memory, percept)** , che integra le nuove percezioni con le informazioni già memorizzate.
  - L'agente, basandosi sulla sua memoria aggiornata, sceglie la **migliore azione possibile**. Questa scelta viene fatta con la funzione **CHOOSE-BEST-ACTION(memory)** . L'agente valuta le possibili azioni in base a ciò che ha appreso e decide l'azione da eseguire.
  - Dopo aver scelto l'azione, l'agente **aggiorna** di nuovo la sua memoria per tenere conto dell'azione eseguita. Questo serve a **ricordare** quali azioni sono state fatte e a monitorare l'effetto delle azioni sull'ambiente.

## 💡 Funzionamento generale:

1. **Percezione:** L'agente riceve nuove informazioni dall'ambiente.
2. **Aggiornamento della memoria:** Le percezioni vengono integrate nella memoria esistente dell'agente.
3. **Decisione:** L'agente sceglie l'azione migliore basata sulla sua memoria e le informazioni attuali.
4. **Esecuzione:** L'agente aggiorna la memoria con l'azione eseguita e poi esegue l'azione.

## Gioco del Wumpus

Ambiente classico nel campo dell'intelligenza artificiale noto come "Wumpus World", un problema tipico usato per illustrare il comportamento di un **agente razionale** in un ambiente con pericoli e ricompense. In questo caso, l'agente ha un **antagonista** (il Wumpus) e altre minacce come le buche (PIT).



- **Griglia (4x4)**: L'ambiente è rappresentato come una griglia in cui l'agente si muove, partendo dalla casella in basso a sinistra (START, riga 1, colonna 1), cercando di raccogliere l'oro situato in una casella della griglia (casella riga 3, colonna 2).

### Elementi dell'ambiente:

1.  **Wumpus (Antagonista)**: Il Wumpus è una creatura ostile che uccide l'agente se quest'ultimo finisce nella stessa casella. Tuttavia, il Wumpus emette un odore ("stench") percepibile nelle caselle adiacenti.
2.  **PIT (Buche)**: Le buche sono trappole in cui l'agente può cadere e morire. Il segno "breeze" indica la presenza di una buca nelle caselle adiacenti.
3.  **Gold (Oro)**: Il tesoro che l'agente deve recuperare. Una volta trovato l'oro, l'obiettivo dell'agente è tornare indietro alla casella di partenza senza cadere in trappole o essere ucciso dal Wumpus.

### Percezioni dell'agente:

L'agente raccoglie **percezioni** dall'ambiente in base a ciò che rileva nelle caselle adiacenti:

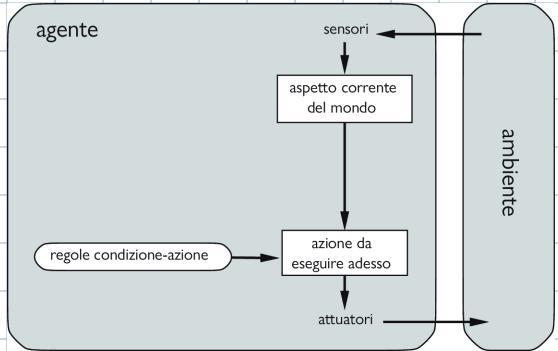
-  **Stench**: Indica che il Wumpus è vicino, quindi l'agente è a una casella di distanza dal Wumpus.
-  **Breeze**: Indica che c'è una buca nelle vicinanze (una delle caselle adiacenti).
-  **Gold**: Quando l'agente si trova nella stessa casella dell'oro, lo percepisce e può prenderlo.

# AGENTI REATTIVI SEMPLICI

- AZIONE SCELTA BASATA SU PERCEZIONE CORRENTE
- IGNORA LA STORIA PERCETTIVA PRECEDENTE.

## REGOLA CONDIZIONE - AZIONE

↳ ESEMPIO: SE LA MACCHINA DANANTI FRENA THEN INIZIA A FRENARE



```

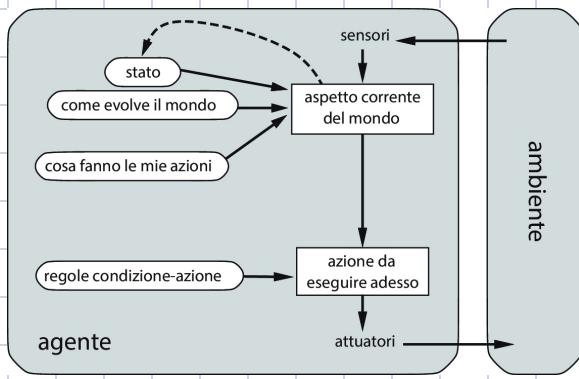
function Agente-Reattivo-Semplice(percezione) returns un'azione
    persistent: regole, un insieme di regole condizione-azione
    stato ← Interpreta-Input(percezione)
    regola ← Regola-Corrispondente(stato, regole)
    azione ← regola.Azione
    return ← azione (es. "suck")

```

# AGENTI BASATI SU MODELLO

UTILIZZA UN MODELLO PER RAPPRESENTARE L'AMBIENTE E PRENDERE DECISIONI PIÙ INFORMATE E PREDIRE LE CONSEGUENZE DELLE SUE AZIONI

↳ ESEMPIO: GIRANDO IL VOLANTE LA MACCHINA SVOLTA IN QUELLA DIREZIONE



```

function Agente-Reattivo-Basato-Su-Modello(percezione) returns un'azione
    persistent:
        stato, la concezione corrente dello stato del mondo da parte dell'agente
        modello_transizione, una descrizione della dipendenza dello stato successivo dallo stato corrente e dall'azione
        modello_sensoriale, una descrizione di come lo stato del mondo attuale è riflesso nelle percezioni dell'agente
        regole, un insieme di regole condizione-azione
        azione, l'azione più recente, inizialmente nessuna
    stato ← Aggiorna-Stato(stato, azione, percezione, modello_transizione, modello_sensoriale)
    regola ← Regola-Corrispondente(stato, regole)
    azione ← regola.Azione
    return ← azione

```

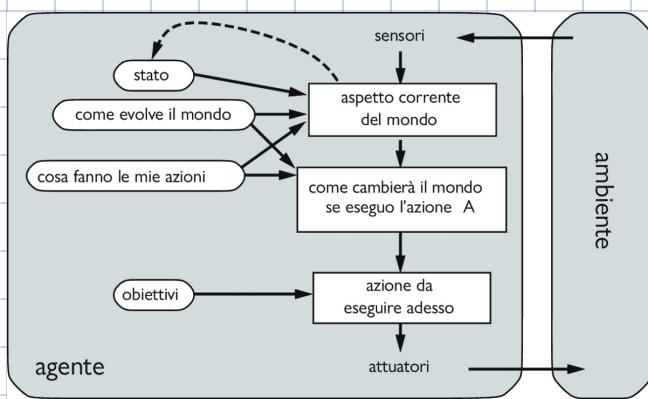
DATO UNO STATO  
REALE, DESCRIUVE COME I SENSORI  
GENERO DATI

Aspetto	Modello di Transizione	Modello Sensoriale
Domanda a cui risponde	"Cosa succederà dopo un'azione?"	"Cosa sto osservando?"
Ruolo	Prevede il nuovo stato in base all'azione	Interpreta le osservazioni per determinare lo stato attuale
Input principale	Stato corrente $S_t$ , azione $A_t$	Stato reale $S_t$
Output principale	Nuovo stato $S_{t+1}$	Osservazione $O_t$
Esempio	Prevedere dove sarà un robot dopo essersi spostato	Interpretare una lettura GPS per sapere dove si trova

# AGENTI BASATI SU OBIETTIVI

UTILIZZANO LO STATO OBIETTIVO (GOAL) PER PRENDERE DECISIONI DA ESEGUIRE, LO SCOPO È QUELLO DI RAGGIUNGERE L'OBBIETTIVO.

→ ESEMPIO: STATO CORRENTE = MACCHINA AD UN INCROCIO, DOBBIAMO DECIDERE DOME SVOLTARE, LO FACCIAMO BASANDOCI SULLA META DA RAGGIUNGERE.



PIÙ FLESSIBILE, LA CONOSCENZA CHE GUIDA LE SUE AZIONI È ESPlicita e può essere modificata.

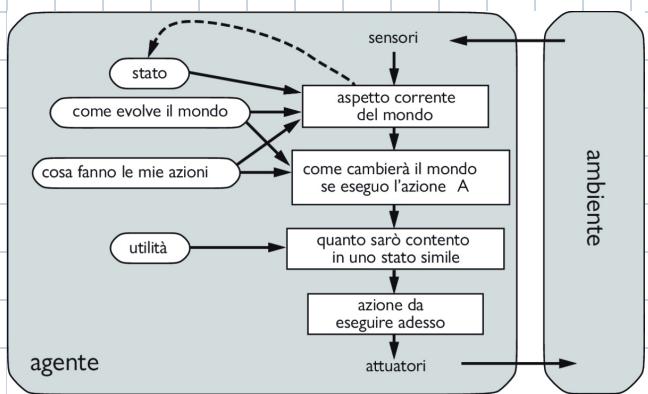
LE REGOLE DI UN AGENTE REATTIVO AD ESEMPIO FUNZIONERANNO ESCUSIVAMENTE PER UNA PRECISA META.

# AGENTI BASATI SU UTILITÀ

CERCA DI MASSIMIZZARE IL VALORE ATTESO IN BASE AD UN FATTORE DI "CONTENTEZZA" ANCHE DETTO UTILITÀ.

→ ESEMPIO: ARRIVARE AD UNA META, IL PERCORSO SCELTO È BUONO?

↓  
UTILIZZA FUNZIONE DI UTILITÀ PER ASSEGNARE UN PUNTEGGIO



OLTE A TENERE TRACCIA DELLO STATO ATTUALE IMPIEGA LA FUNZIONE DI UTILITÀ PER MISURARE LE SUE PREFERENZE TRA I VARI STATI DEL MONDO.

# AGENTI BASATI SU APPRENDIMENTO

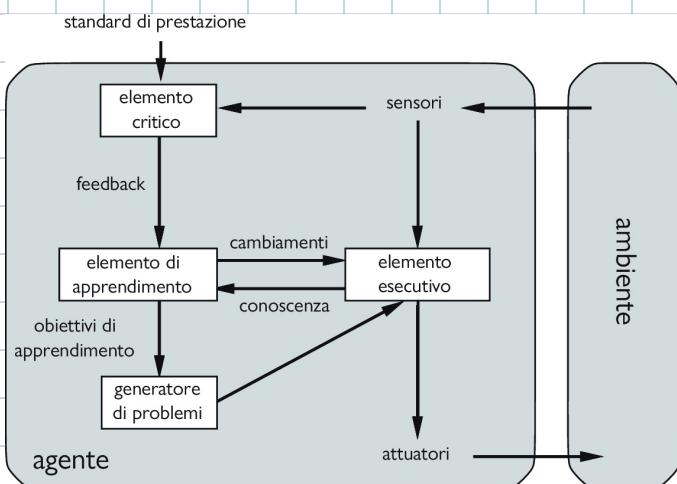
AGENTE CAPACE DI APPRENDERE, PUÒ OPERARE IN AMBIENTI INIZIALMENTE SCONOSCIUTI POI DIVENTANDO SEMPRE PIÙ ESPERTI.

↳ GLI AGENTI SOPRACITATI POSSONO ESSERE COSTITUITI TRAMITE AGENTI DI QUESTO TIPO.

SUDDIVISO IN 4 COMPONENTI ASTRATTI:

- ELEMENTO DI APPRENDIMENTO → RESPONSABILE MIGLIORAMENTO INTERNO
- ELEMENTO ESECUTIVO → RESPONSABILE DELLA SELEZIONE DELLE AZIONI ESTERNE
- ELEMENTO CRITICO → RICORDA LE PRESTAZIONI DELL'AGENTE DETERMINANTE SE E COME MODIFICARE L'ELEM. ESECUTIVO AFFINCHÉ SI COMPORTI MEGLIO IN FUTURO.
- GENERATORE DI PROBLEMI → LO SCOPO È SUGGERIRE NUOVE AZIONI CHE PORTINO A ESPERIENZE MAI AFFRONTATE, AUMENTANDO COSÌ LA CAPACITÀ DI GENERALIZZAZIONE DELL'AMBIENTE.

LA DECISIONE NON È QUINDI UNA CONSEGUENZA DIRETTA DELL'UTILITÀ MA LE DECISIONI VENGONO MIGLIorate IN BASE AI FEEDBACK RICEVUTI E L'APPRENDIMENTO.



Per esempio, supponiamo che l'agente tassista non riceva mai alcuna mancia dai passeggeri che sono stati rozzamente sbalzati per tutto il viaggio. Lo standard di prestazione esterno deve informare l'agente che la perdita di tutte le manciate rappresenta un contributo negativo alla sua performance globale; l'agente potrebbe allora apprendere che la guida nervosa non contribuisce alla sua utilità. In un certo senso, lo standard di prestazione caratterizza una parte delle percezioni come **ricompense** (o rispettivamente **penalità**) che rappresentano un feedback diretto sulla qualità del comportamento dell'agente. In questo modo si possono descrivere standard di prestazione innati, quali il dolore e la fame negli animali.

# Tipi di Agenti:



## Agente basato su modello

Un **agente basato su modello** non si limita a reagire immediatamente alle percezioni, ma utilizza una rappresentazione interna dello stato del mondo (il modello) per prendere decisioni più **informate** e ragionate.

### ☰ Struttura

1. **Guida delle azioni:** Si basano su percezioni e regole condizione-azione, reagiscono in base allo stato attuale.
2. **Obiettivo:** Non hanno un obiettivo esplicito.
3. **Pianificazione:** Non pianificano a lungo termine, prendono decisioni immediate.
4. **Flessibilità:** Limitati a regole predefinite, meno flessibili.
5. **Decisione basata su utilità:** Non considerano l'utilità, eseguono le azioni in base alle regole.
6. **Apprendimento:** Nessun apprendimento, reagiscono solo alle percezioni e regole predefinite.
7. **Feedback e miglioramento:** Non ricevono feedback, non migliorano nel tempo.

## Strategia dell'agente:

- **Esplorazione prudente:** L'agente deve muoversi esplorando una casella alla volta e valutare le percezioni ricevute. Ad esempio, se rileva un odore (stench), sa che il Wumpus è nelle vicinanze e dovrebbe evitare di entrare in quella casella.
- **Evitare rischi:** Basandosi sulle percezioni di breeze, l'agente sa dove potrebbero essere le buche, quindi deve pianificare il percorso in modo da evitare di entrare in quelle caselle.
- **Uccidere il Wumpus:** Se l'agente ha un'arma (ad esempio, una freccia), potrebbe decidere di uccidere il Wumpus per rendere l'ambiente più sicuro. Tuttavia, questo dipende dalla strategia dell'agente e dalla situazione.

## Obiettivo dell'agente:

L'agente deve navigare attraverso l'ambiente, evitando i pericoli (Wumpus e buche), cercando di trovare e raccogliere l'oro e poi tornare al punto di partenza. L'agente deve essere **razionale**, utilizzando le sue percezioni (come stench e breeze) per evitare i pericoli e scegliere il miglior percorso possibile.

# Agenti con obiettivo

Questi agenti sono più avanzati rispetto agli agenti reattivi o basati su modello, in quanto non solo reagiscono a ciò che percepiscono, ma pianificano e selezionano le loro azioni in funzione di un **obiettivo** specifico.

## ☰ Struttura

1. **Guida delle azioni:** Sceglie azioni che portano a un obiettivo specifico.
2. **Obiettivo:** Guidati da obiettivi chiari e definiti.
3. **Pianificazione:** Pianificano una sequenza di azioni per raggiungere l'obiettivo.
4. **Flessibilità:** Più flessibili rispetto agli agenti basati su modello grazie alla pianificazione.
5. **Decisione basata su utilità:** Sceglie azioni in base all'obiettivo finale, ma non valuta la qualità degli stati intermedi.
6. **Apprendimento:** Non apprendono, ma possono adattare le azioni a seconda degli stati successivi.
7. **Feedback e miglioramento:** Non ricevono feedback esterno, seguono le regole predefinite.

## Agenti con valutazione di utilità

Gli **agenti con valutazione di utilità** sono una versione avanzata degli agenti con obiettivo. La principale differenza è che non si limitano a scegliere un'azione per raggiungere un obiettivo, ma valutano **quale stato futuro** è preferibile in base a una funzione di **utilità**, che misura il grado di "soddisfazione" o "felicità" dell'agente nello stato futuro.

### ☰ Struttura

1. **Guida delle azioni:** Sceglie azioni che massimizzano la **utilità**, valutando la "felicità" rispetto agli stati futuri.
2. **Obiettivo:** Misurano e massimizzano la **soddisfazione** o utilità nello stato futuro.
3. **Pianificazione:** Pianificano una sequenza di azioni che massimizza l'utilità.
4. **Flessibilità:** Molto flessibili, in grado di valutare e fare compromessi tra azioni che portano a stati diversi..
5. **Decisione basata su utilità:** Valutano ogni stato intermedio e futuro in termini di utilità, preferendo quelli che massimizzano la "felicità".
6. **Apprendimento:** Non apprendono, ma valutano e preferiscono stati con utilità maggiore.
7. **Feedback e miglioramento:** Non c'è un sistema di feedback esplicito, valutano solo stati futuri.

## Agenti che apprendono

Gli **agenti che apprendono** rappresentano un'ulteriore evoluzione rispetto agli agenti basati su modello e agli agenti con obiettivo. Questi agenti sono capaci di **migliorare il loro comportamento** con l'esperienza e di **adattarsi** a nuovi ambienti o cambiamenti nell'ambiente, modificando le loro strategie basate sul feedback.

### ☰ Struttura

1. **Guida delle azioni:** Sceglie azioni in base al feedback ricevuto e apprende dalle esperienze precedenti.
2. **Obiettivo:** Imparano nuovi obiettivi e regole in base al feedback e ai risultati ottenuti.
3. **Pianificazione:** L'apprendimento migliora la capacità di pianificare azioni basate su esperienze passate.
4. **Flessibilità:** Altamente flessibili, migliorano con l'apprendimento continuo e possono adattarsi a nuovi scenari.
5. **Decisione basata su utilità:** Non usano utilità diretta, ma migliorano le decisioni basate su feedback e apprendimento.
6. **Apprendimento:** Sono capaci di apprendere dalle esperienze passate, migliorando le prestazioni e adattandosi.
7. **Feedback e miglioramento:** Basati su feedback continuo, migliorano il comportamento grazie al critico e all'elemento di apprendimento.