

# Agenti basati su conoscenza

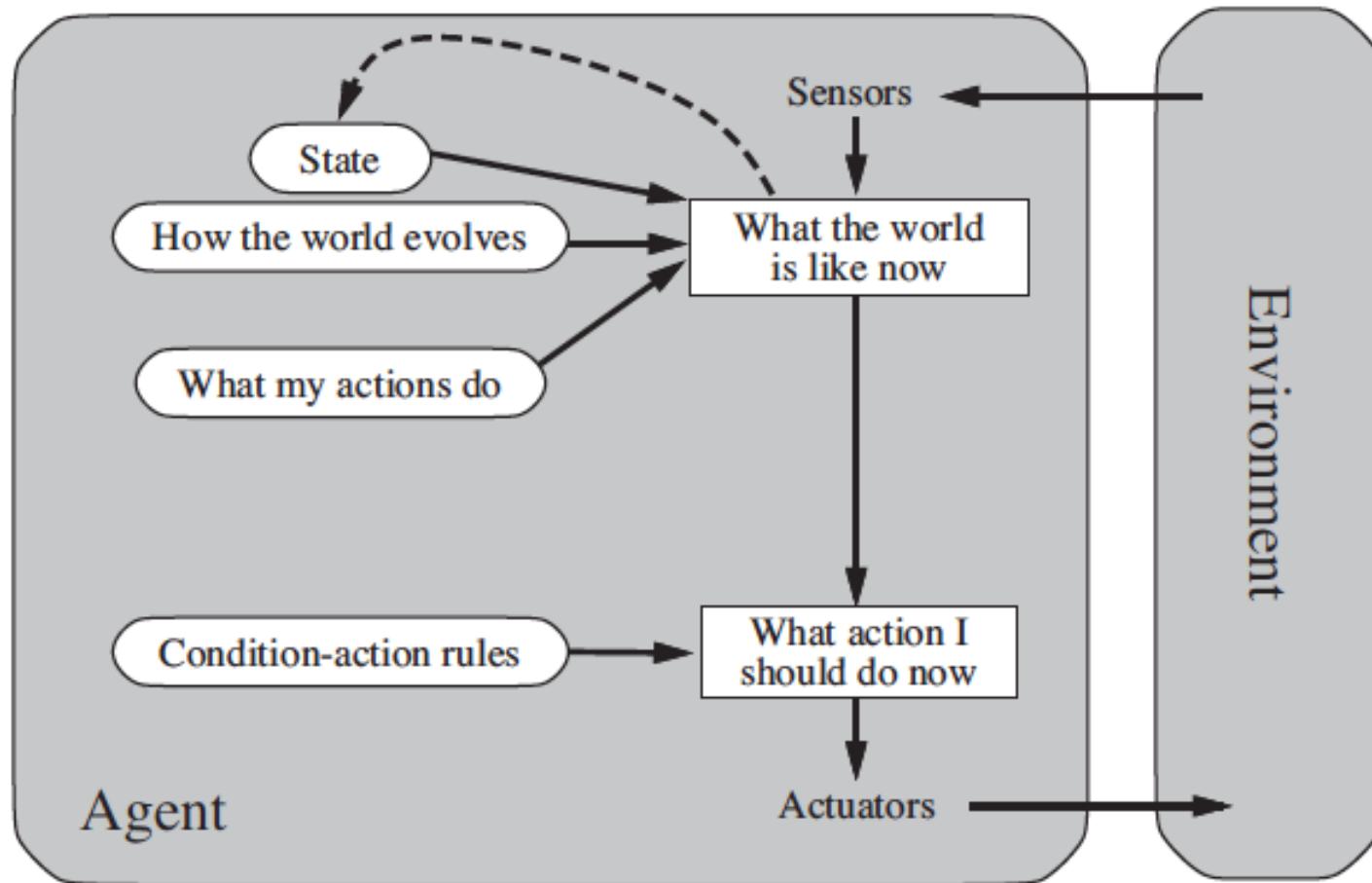
*Come costruire agenti dotati di  
capacità inferenziali*

Paolo Mancarella, 2022/2023  
*paolo.mancarella@unipi.it*

# Di cosa parleremo...

- Avete visto:
  - agenti con stato e con obiettivo in mondi osservabili con stati atomici e azioni descrivibili in maniera semplice
  - enfasi sul processo di ricerca
- Come migliorare le **capacità di ragionamento** degli agenti, dotandoli di rappresentazioni di mondi complessi e **astratti**, non descrivibili semplicemente
- Agenti **basati su conoscenza**, dotati di una KB (*Knowledge Base*) con conoscenza espressa in maniera esplicita e dichiarativa

# Agenti basati su modello



# Agenti “Knowledge Based”

- La maggior parte dei problemi di I.A. sono “*knowledge intensive*”
- Il mondo è tipicamente complesso: ci serve una rappresentazione *parziale* e *incompleta* (un’astrazione) del mondo funzionale agli scopi dell’agente
- Per ambienti parzialmente osservabili e complessi ci servono linguaggi di rappresentazione della conoscenza *più espressivi* e *capacità inferenziali*
- La conoscenza può essere codificata a mano ma anche estratta dai testi o appresa dall’esperienza o estratta/elicitata dagli esperti..

# Approccio dichiarativo vs procedurale

- La KB racchiude tutta la conoscenza necessaria a decidere l'azione da compiere in forma *dichiarativa*
- L'alternativa (*approccio procedurale*) è scrivere un programma che implementa il processo decisionale, una volta per tutte.
- Un agente KB è più flessibile: più semplice acquisire conoscenza incrementalmente e modificare il comportamento con l'esperienza

# Il mondo del Wumpus: un esempio

4	Fetore	Brezza	POZZO
3	Fetore Oro	Brezza Fetore	POZZO
2	Fetore	Brezza	
1	START	Brezza	POZZO
	1	2	3

- *Misura delle prestazioni:*
  - +1000 se trova l'oro, torna in [1,1] e esce;
  - -1000 se muore;
  - -1 per ogni azione;
  - -10 se usa la freccia.
- *Percezioni:*
  - *fetore* nelle caselle adiacenti al Wumpus;
  - *brezza* nelle caselle adiacenti ai pozzi;
  - *luccichio* nelle caselle con l'oro;
  - *bump* se sbatte in un muro;
  - *urlo* se il Wumpus viene ucciso.
  - L'agente non percepisce la sua locazione.
- *Azioni:*
  - *avanti*
  - *a destra* di 90°, *a sinistra* di 90°
  - *afferra* un oggetto
  - *scaglia la freccia* (solo una)
  - *Esce*
- Ambienti generati a caso ([1,1] safe)

# Il mondo del Wumpus: uno scenario (1)

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
A			
OK	OK		

(a)

- Le percezioni sono una quintupla [*Fetore, Brezza, Luccichio, Bump, Urlo*]
- Percezione in [1,1]  
[*none,none,none,none,none*]
- Né *Brezza* né *Fetore* in [1,1], quindi [1,2] e [2,1] sono sicure (**OK**)
- L'agente decide di spostarsi in [2,1] ...

## Il mondo del Wumpus: uno scenario (2)

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1	2,1 A B OK	3,1 P?	4,1

(b)

- Percezione in [2,1]  
[none,Brezza,none,none,none]
- L'agente percepisce una *Brezza*.  
Quindi c'è un *pozzo* in [2,2] o [3,1] (**P?**)
- L'agente torna in [1,1] e poi si sposta in [1,2] ...

## Il mondo del Wumpus: uno scenario (3)

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A F OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(a)

- Percezione in [2,1]  
[Fetore,none,none,none,none]
- Il Wumpus non può essere in [1,1] (*safe* per definizione), né in [2,2] (altrimenti avrebbe percepito *Fetore* anche in [2,1]). Quindi è in [1,3].
- Siccome non c'è *Brezza* in [1,2], [2,2] è OK e ci deve essere un *Pozzo* in [3,1].

# Il mondo del Wumpus: uno scenario (4)

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A F L B	3,3 P?	4,3
1,2 F V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(b)

- L'agente si sposta in [2,2] e poi in [2,3]
- Percezione in [2,3]  
*[Fetore, Brezza, Luccichio, none,none]*
- Percepisce un *Luccichio*, afferra l'*Oro* e torna sui suoi passi, percorrendo caselle OK.

# Agente basato su conoscenza

- Un agente basato su conoscenza mantiene una *base di conoscenza* (KB): un insieme di *enunciati* espressi in un linguaggio di rappresentazione
- Interagisce con la KB mediante una interfaccia funzionale *Tell-Ask*:
  - *Tell*: per aggiungere nuovi enunciati a KB
  - *Ask*: per interrogare la KB
  - *Retract*: per eliminare enunciati
- Gli enunciati nella KB rappresentano le *opinioni/credenze dell'agente*
- Le risposte  $\alpha$  devono essere tali che  $\alpha$  *discende necessariamente* dalla KB

# Il problema fondamentale da risolvere

- *Il problema:* data una base di conoscenza KB, contenente una rappresentazione dei fatti che si **ritengono veri**, come dedurre che un certo fatto  $\alpha$  è vero di conseguenza?

$KB \models \alpha$

*(conseguenza logica)*

# Programma di un agente KB

```
function Agente-KB (percezione) returns un'azione
    persistent: KB, una base di conoscenza
                t, un contatore, inizialmente a 0, che indica il tempo
    TELL(KB, Costruisci-Formula-Percezione(percezione, t ))
    azione ← ASK(KB, Costruisci-Query-Azione( t ))
    TELL(KB, Costruisci-Formula-Azione(azione, t ))
    t ← t + 1
    return azione
```

# Base di conoscenza vs base di dati

- *Base di conoscenza*: una rappresentazione esplicita, parziale e compatta, in un *linguaggio simbolico*, che contiene:
  - fatti di tipo specifico
    - ✓ *Casella [1, 1] ok*
    - ✓ *C'è un Pozzo in [3,1]*
  - fatti di tipo generale, o regole
    - ✓ *C'è brezza nelle caselle adiacenti ai pozzi*
- *Base di dati*: solo fatti specifici, solo recupero (*retrieval*)
- Quello che caratterizza una KB è *la capacità inferenziale*
  - derivare nuovi fatti da quelli memorizzati esplicitamente

Es. *C'è un Pozzo in [3, 1]*

*Il Wumpus è in [1,3]*

## Il *trade-off* fondamentale della R.C.

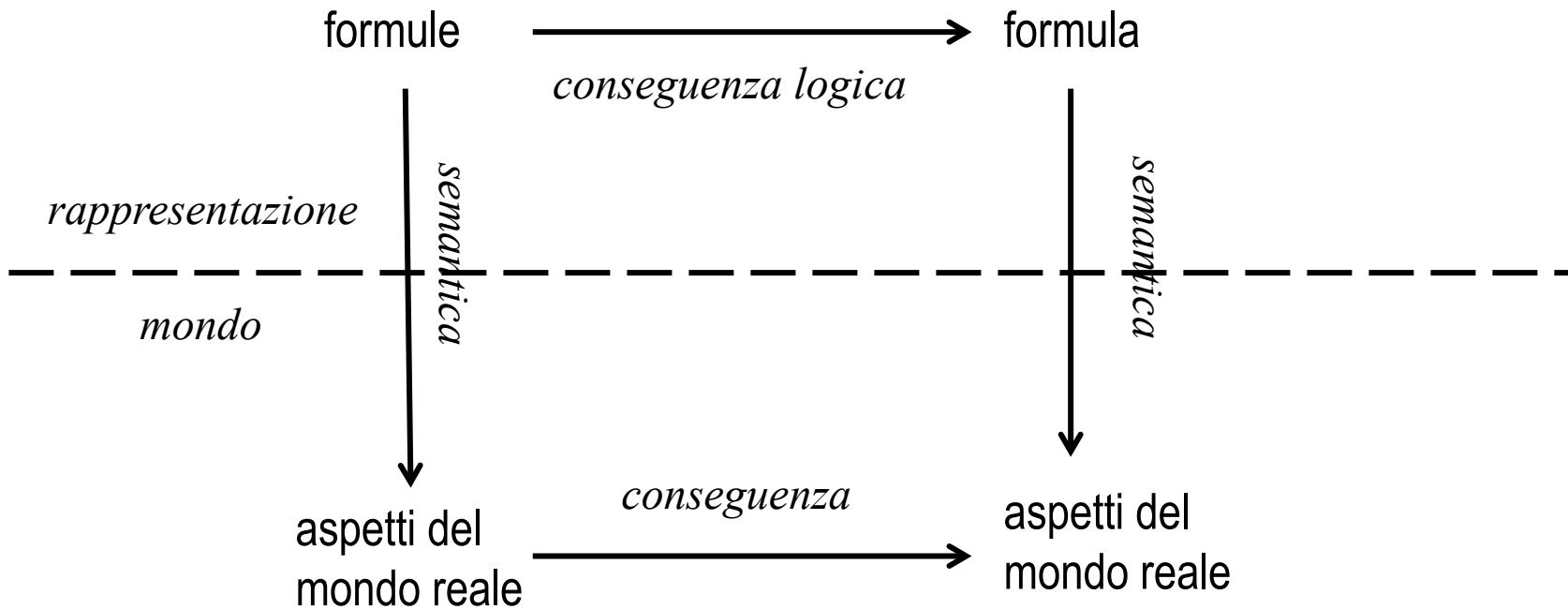
- Sfortunatamente più il linguaggio è *espressivo*, meno *efficiente* è il meccanismo inferenziale.
- Il problema ‘fondamentale’ in R.C. è trovare il giusto compromesso tra:
  - *Espressività* del linguaggio di rappresentazione
  - *Complessità* del meccanismo inferenziale
- Questi due obiettivi sono in contrasto e si tratta di mediare tra queste due esigenze

# Formalismi per la Rappresentazione della Conoscenza

Un formalismo per la rappresentazione della conoscenza ha tre componenti:

1. Una *sintassi*: un linguaggio composto da un vocabolario e regole per la formazione delle frasi (*enunciati*)
2. Una *semantica*: stabilisce una corrispondenza tra gli enunciati e i fatti del mondo; se un agente ha un enunciato  $\alpha$  nella sua KB, crede che il fatto corrispondente sia vero nel mondo
3. Un *meccanismo inferenziale* (codificato o meno tramite regole di inferenza come nella logica) che ci consente di inferire nuovi fatti.

# Rappresentazione e mondo



# Logica come linguaggio per la R.C.

- Qual è la complessità computazionale del problema  $\text{KB} \models \alpha$  nei vari linguaggi logici?
- Quali sono gli algoritmi di decisione e le strategie di ottimizzazione?
- I linguaggi logici:
  - calcolo proposizionale (PROP)
  - logica dei predicati del prim'ordine (FOL)
- Sono adatti per la rappresentazione della conoscenza?

# Agenti logici: calcolo proposizionale

*Paolo Mancarella*  
a.a. 2022/2023

# Sintassi

- La sintassi definisce quali sono le frasi legittime (**ben formate**) del linguaggio:

$$formula \rightarrow formulaAtomica \mid formulaComplessa$$
$$formulaAtomica \rightarrow True \mid False \mid simbolo$$
$$simbolo \rightarrow P \mid Q \mid R \mid \dots$$
$$formulaComplessa \rightarrow \neg formula$$
$$\mid (formula \wedge formula)$$
$$\mid (formula \vee formula)$$
$$\mid (formula \Rightarrow formula)$$
$$\mid (formula \Leftrightarrow formula)$$

# Sintassi: esempi

- $((A \wedge B) \Rightarrow C)$

Possiamo omettere le parentesi assumendo questa **precedenza** tra gli operatori:

$\neg > \wedge > \vee > \Rightarrow, \Leftrightarrow$

- $\neg P \vee Q \wedge R \Rightarrow S$  è la stessa cosa di  $((\neg P) \vee (Q \wedge R)) \Rightarrow S$

- Es. dal mondo del Wumpus

- $P_{1,1}$  c'è un pozzo in [1,1]

- $W_{2,3}$  il Wumpus è in [2,3]

# Semantica e mondi possibili (modelli)

- La semantica ha a che fare col significato delle frasi: definisce se un enunciato è vero o falso rispetto a una *interpretazione* (mondo possibile)
- Una interpretazione definisce un valore di verità per tutti i simboli proposizionali.
  - Esempio:  $\{P_{1,1} \text{ vero}, P_{1,2} \text{ falso}, W_{2,3} \text{ vero}\}$
- Il valore di una formula complessa è fissato di conseguenza
  - $P_{1,1} \Rightarrow W_{2,3} \vee P_{1,2}$  è *vera* in questa interpretazione
- Un *modello* di una formula (resp. insieme di formule) è una interpretazione *che la rende vera* (resp. che rende vere *tutte* le formule dell'insieme)

# Semantica compositiva

- Il significato di una frase è determinato dal significato dei suoi componenti, a partire dalle frasi atomiche (i simboli proposizionali)
  - *True* sempre vero; *False* sempre falso
  - $P \wedge Q$ , vero se  $P$  e  $Q$  sono entrambi veri
  - $P \vee Q$ , vero se  $P$  oppure  $Q$ , o entrambi, sono veri
  - $\neg P$  vero se  $P$  è falso
  - $P \Rightarrow Q$ , vero se  $P$  è falso oppure  $Q$  è vero
  - $P \Leftrightarrow Q$ , vero se entrambi veri o entrambi falsi

# Conseguenza logica

- Una formula  $\alpha$  è *conseguenza logica* di un insieme di formule  $KB$  se e solo se in ogni modello di  $KB$ , anche  $\alpha$  è vera ( $KB \models \alpha$ )
- Indicando con  $M(KB)$  i modelli dell'insieme di formule in  $KB$  e con  $M(\alpha)$  l'insieme delle interpretazioni che rendono  $\alpha$  vera (i **modelli** di  $\alpha$ ) ...

$$KB \models \alpha \text{ sse } M(KB) \subseteq M(\alpha)$$

# Esempio dal mondo del Wumpus (1)

- La KB iniziale  $KB_0$  è costituita dalle regole generali del WW e dal fatto che la casella iniziale è *safe* per definizione

$$\neg W_{1,1} \quad \neg P_{1,1}$$

$$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

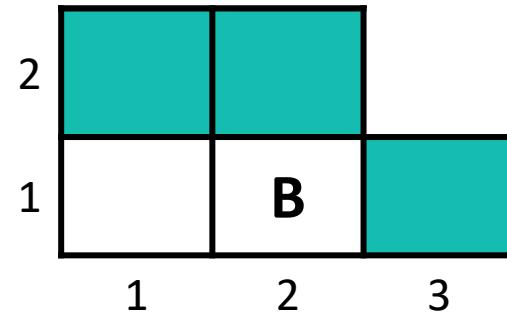
$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

...

La casella iniziale è *safe*

esempi di regole generali

## Esempio dal mondo del Wumpus (2)



L'agente non ha percepito nulla in [1,1] e si è spostato in [2,1] dove ha percepito una *Brezza*

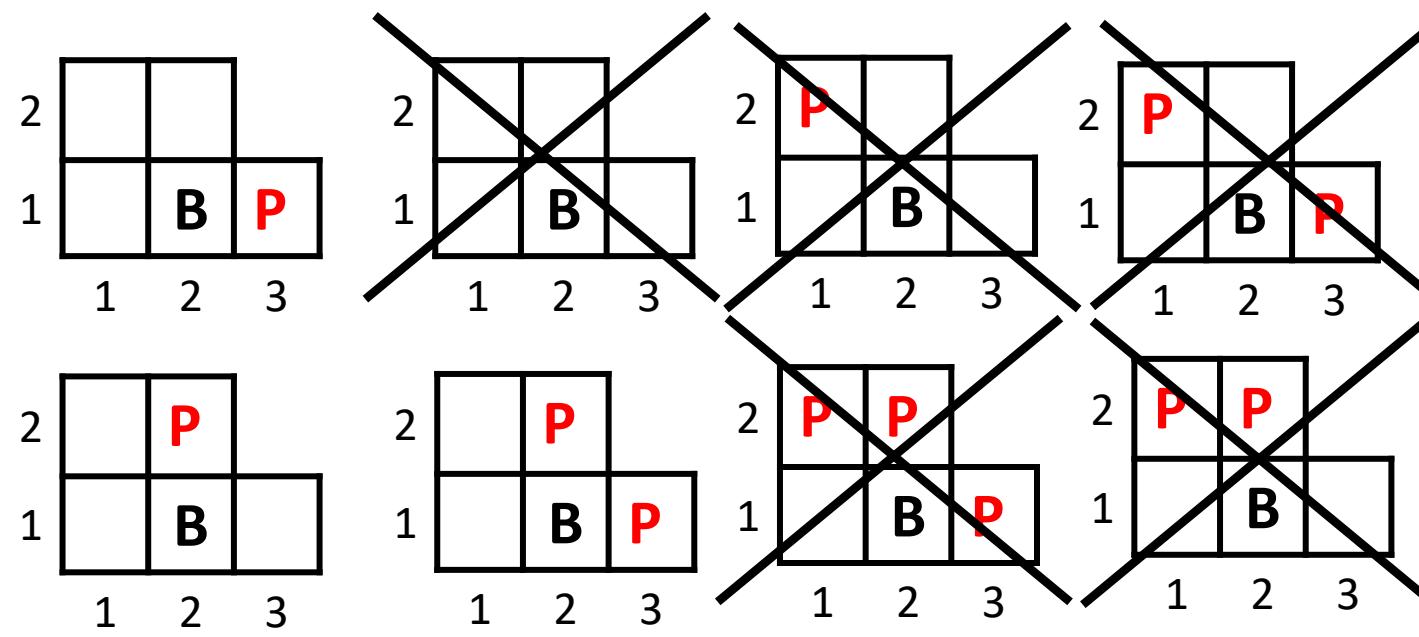
- $KB_1$  a questo punto è  $KB_0$  più i *fatti* corrispondenti alle percezioni  
 $KB_0 \cup \{\neg B_{1,1}, B_{2,1}, \neg F_{1,1}, \neg F_{2,1}, \dots\}$
- Domandiamoci se le stanze adiacenti non contengono *Pozzi*

$$KB_1 \models \neg P_{1,2}$$
$$KB_1 \models \neg P_{2,2}$$
$$KB_1 \models \neg P_{3,1}$$



# Esempio dal mondo del Wumpus (3)

- Considerando solo l'esistenza di pozzi nelle 3 caselle  $P_{1,2}$ ,  $P_{2,2}$  e  $P_{3,1}$ , ci sono  $2^3=8$  possibili interpretazioni o mondi possibili



$$KB_1 \models \neg P_{1,2}$$

$$KB_1 \models P_{2,2} \vee P_{3,1}$$

Nei modelli di  $KB_1$  nulla  
possiamo concludere su  
 $\neg P_{2,2}$  e  $\neg P_{3,1}$   
(né su  $P_{2,2}$  e  $P_{3,1}$  ...)