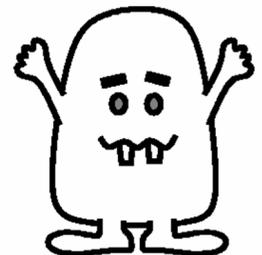


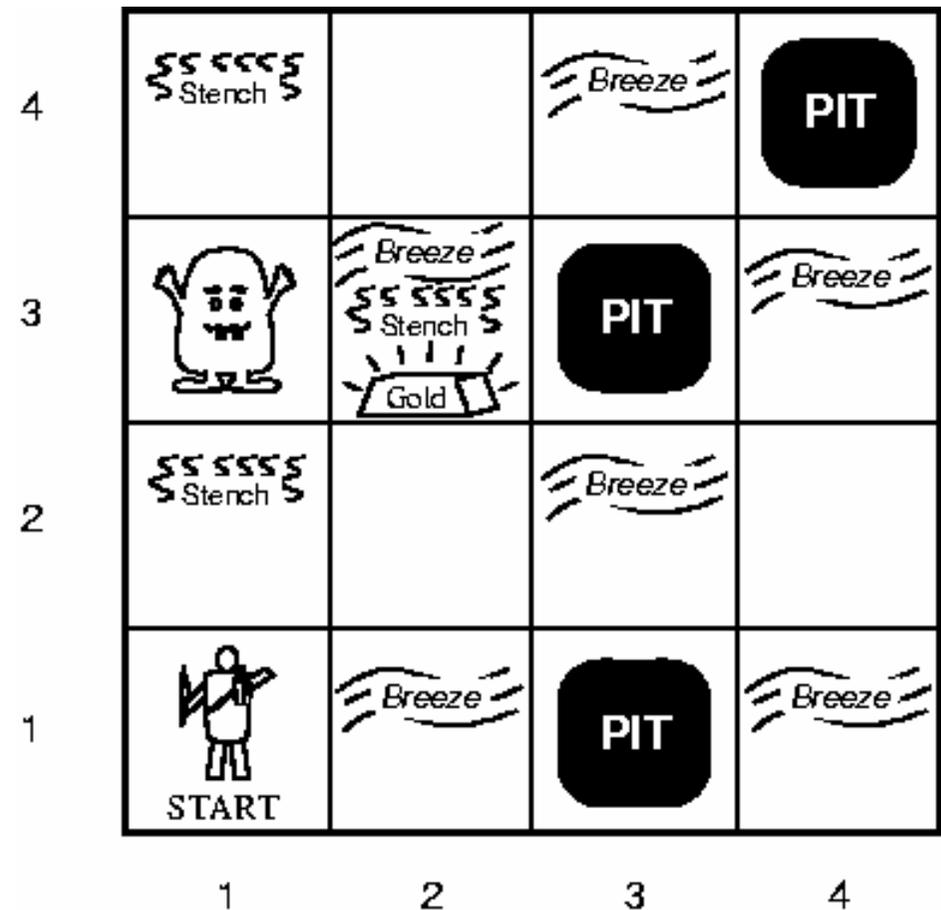
# Intelligenza Artificiale I

## Esercitazione 5 Il mondo di Wumpus

Marco Piastra



# Il mondo



## ▪ Ambiente

Una caverna: le caselle di un reticolo rettangolare (*Cave*)

Un accesso - START: (*Exit*)

Alcune caselle non accessibili (*Nocave*)

Alcune caselle sono una trappola (*Pit*)

Dalle trappole esce vento (*Breeze*)

## Personaggi e scopo

- **Wumpus**

Mostro della caverna

Letale: chi lo incontra muore

Piuttosto fetente (*Stench*)

- **Oro (*Gold*)**

Si trova in una sola casella ed emette un bagliore (*Glitter*)

Si può afferrare solo nella casella che lo contiene

- **Cacciatore (*Hunter*)**

Umano, quindi mortale:

muore se cade in una trappola o entra in una casella con il Wumpus

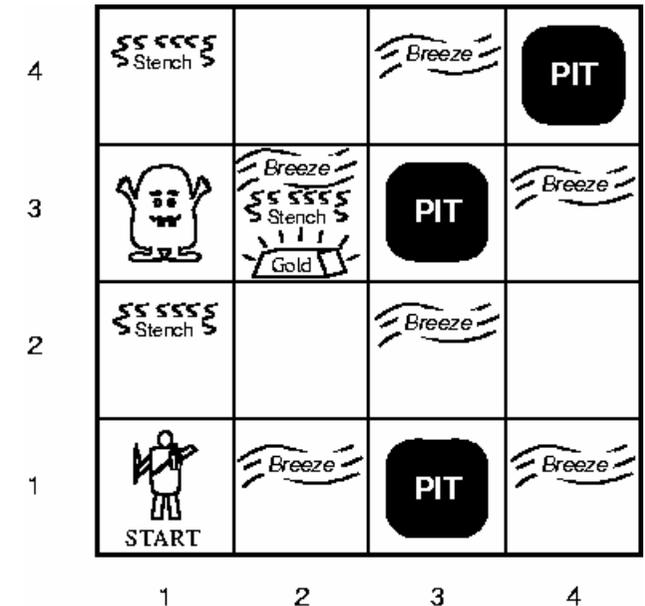
Azioni possibili:

Movimento di una casella alla volta (*Move*)

Afferra un pezzo d'oro (*Pick-up*)

- **Goal**

Il cacciatore deve prendere l'oro e uscire, vivo



# Agente ragionatore

Il cacciatore, ciclo iterativo di fasi

- **Percezione (*Sense*)**

Percepisce la brezza, il bagliore, il fetore

- **Pensiero (*Think*)**

Ragiona e deriva nuovi fatti dalle percezioni (*Evaluate*)

Mantiene una propria rappresentazione del mondo esterno (*Belief*)

Desidera obiettivi (*Desire*)

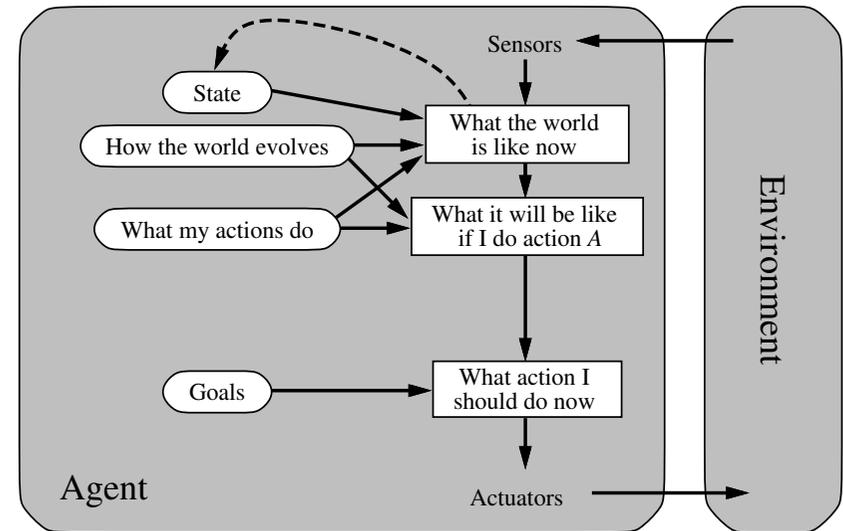
- **Pianificazione (*Plan*)**

Sceglie un obiettivo tra quelli desiderati (*Intention*)

Lo assume come scopo (*Goal*)

- **Azione (*Act*)**

Esegue azioni adeguate allo scopo



## Esempio: il cacciatore razionale

OK			
OK A	OK		

Il cacciatore si trova nella posizione iniziale

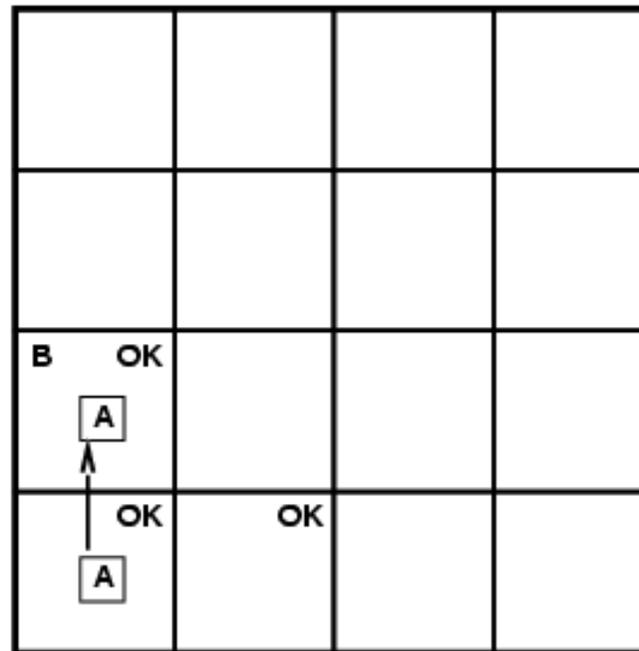
Non c'è brezza, bagliore o fetore (*Sense*)

Quindi le caselle adiacenti sono OK (*Think*)

Essendo alla ricerca dell'oro, desidera esplorare (*Desire*)

Intende muoversi in una casella adiacente (*Plan*)

## Esempio: il cacciatore razionale

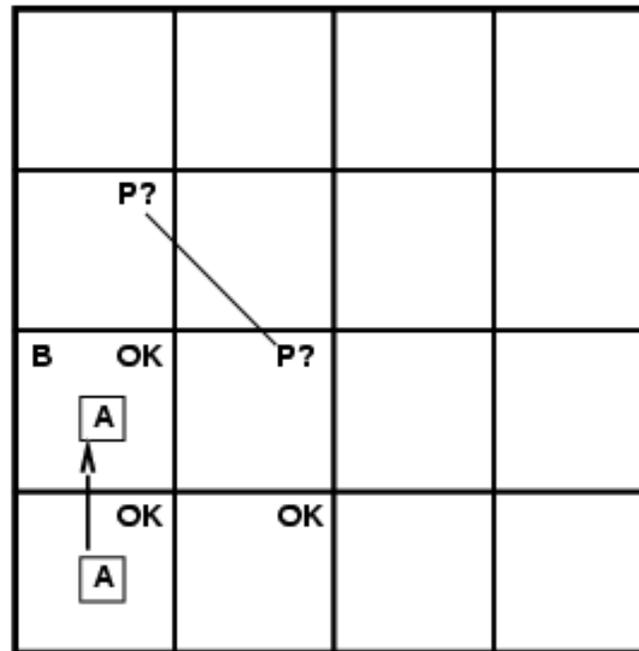


Il cacciatore si muove di una casella (*Act*)

Sente brezza (*Sense*)

Quindi?

## Esempio: il cacciatore razionale



Il cacciatore si muove di una casella (*Act*)

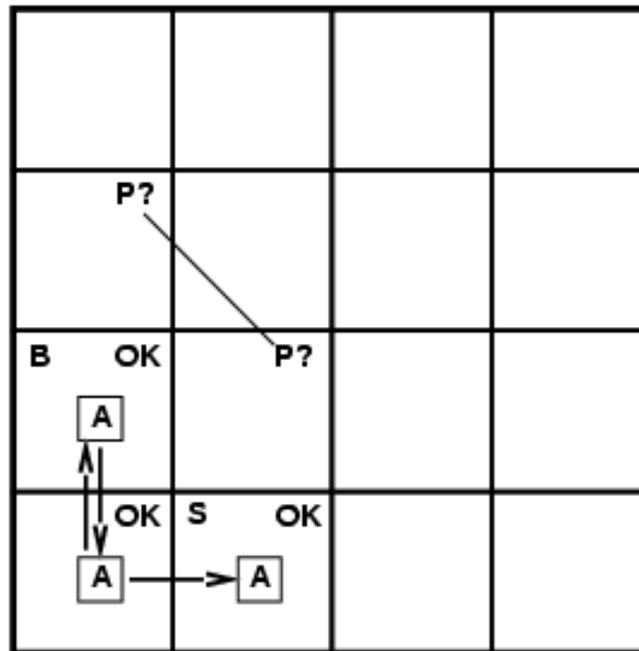
Sente brezza (*Sense*)

Quindi un trappola si trova in una delle caselle adiacenti non esplorate (*Think*)

Due ipotesi, a questo punto

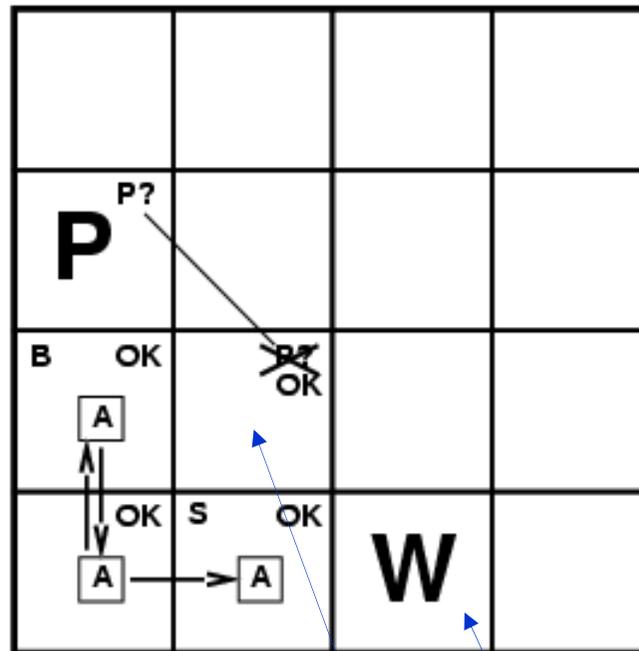
Meglio tornare indietro ed esplorare l'altra casella OK (*Plan*)

## Esempio: il cacciatore razionale



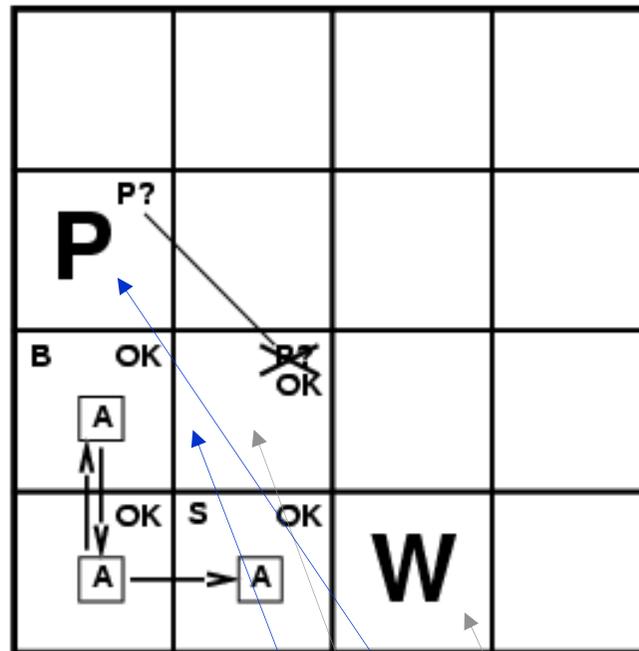
Il cacciatore torna indietro ed esplora l'altra casella (*Act*)  
 Sente il fetore (*Sense*)  
 Quindi?

## Esempio: il cacciatore razionale



Il cacciatore torna indietro ed esplora l'altra casella OK (*Act*)  
 Sente il fetore (*Sense*)  
 Quindi il Wumpus non può essere qui ma è qui (*Think*)

## Esempio: il cacciatore razionale



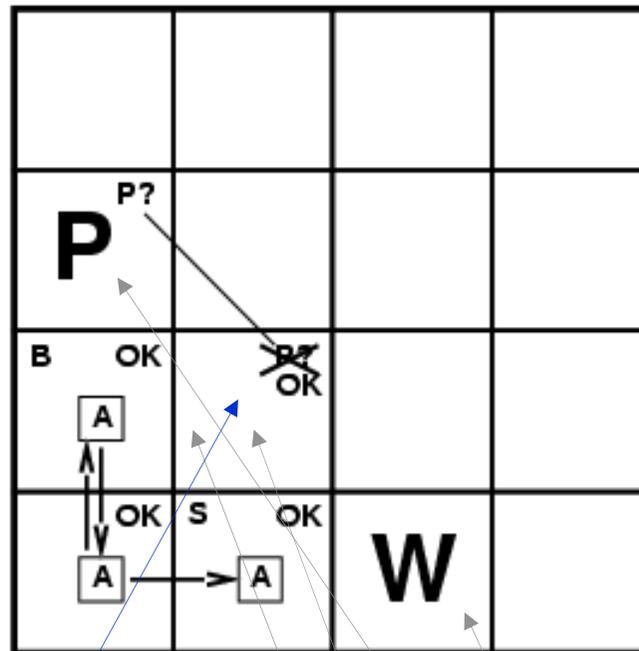
Il cacciatore torna indietro ed esplora l'altra casella OK (*Act*)

Sente il fetore (*Sense*)

Quindi il Wumpus non può essere qui ma è qui (*Think*)

Quindi la trappola non può essere qui ma è qui (*Think*)

## Esempio: il cacciatore razionale



Il cacciatore torna indietro ed esplora l'altra casella OK (*Act*)

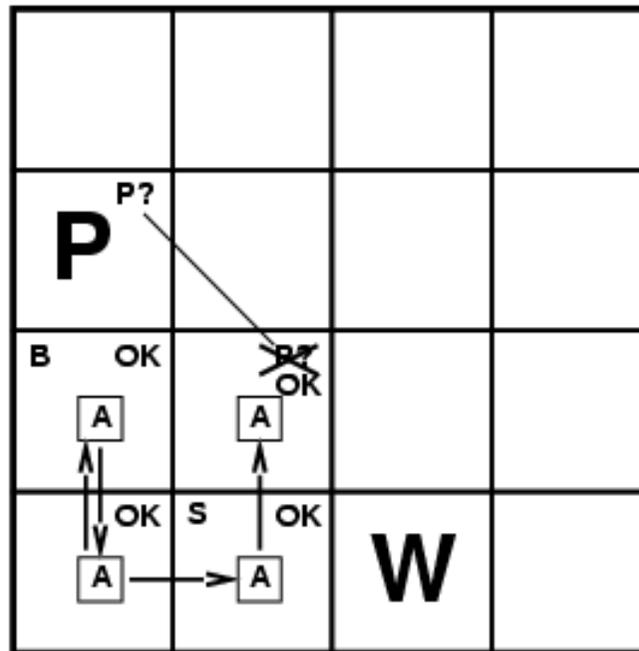
Sente il fetore (*Sense*)

Quindi il Wumpus non può essere qui ma è qui (*Think*)

Quindi la trappola non può essere qui ma è qui (*Think*)

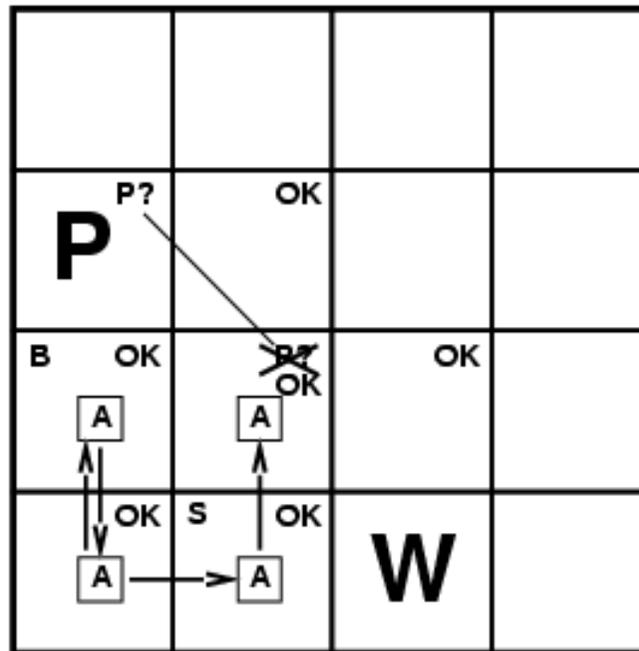
Quindi questa casella è OK (*Think*)

## Esempio: il cacciatore razionale



Il cacciatore esplora la nuova casella OK (*Act*)  
 Non sente nulla (*Sense*)  
 Quindi?

## Esempio: il cacciatore razionale



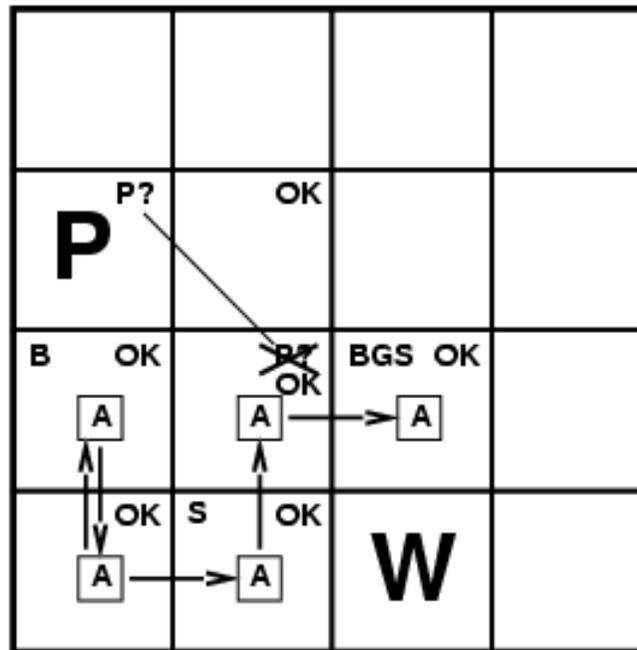
Il cacciatore esplora la nuova casella OK (*Act*)

Non sente nulla (*Sense*)

Quindi le due caselle adiacenti sono OK (*Think*)

Intende muoversi in una casella adiacente (*Plan*)

## Esempio: il cacciatore razionale



Il cacciatore esplora la casella adiacente (*Act*)

Sente brezza, vede il bagliore, sente il fetore (*Sense*)

Quindi nella casella c'è l'oro (*Think*)

Desidera prendere l'oro e poi andarsene (*Desire*)

# Il cacciatore in Jess

## ▪ Genesi e simulazione dell'ambiente

### GENESIS (poco interessante)

file **cave0.jess** : descrizione del mondo

dimensioni, posizione del wumpus, trappole, dov'è l'oro, uscita, nome dell'agente

**buildworld** : costruisce il mondo come un grafo a reticolo,

i nodi (impliciti) sono le coppie  $\langle x,y \rangle$ , gli archi sono descritti come **(adj ?x1 ?y1 ?x2 ?y2)**

**retract-nocaves**: si eliminano gli archi **adj** per i nodi non accessibili (roccia)

**put-hunter-in-caves**: si mette il cacciatore all'ingresso (**exit**)

La descrizione del mondo NON deve essere accessibile al cacciatore:  
deve costruirsi la propria rappresentazione

### SIMULATE

Le regole di simulazione dell'ambiente accedono alla descrizione del mondo e possono modificarla

- Cadere in una trappola
- Incontrare il wumpus
- Movimento del wumpus (non accade in questa versione: il wumpus è immobile)

# Il cacciatore in Jess

## ▪ SENSE

Rappresentazione interna: l'insieme dei fatti **cave** (vedere `deftemplate`)  
Posizione del nodo nel grafo + attributi (conoscenze del cacciatore)

Il mondo visto dal cacciatore

**enter-new-cave, enter-cave-for-first-time, notice-other-caves**

Aggiungi un fatto **cave** per ciascun nodo del grafo che diviene noto

Acquisizione dei dati percettivi

**sense-breeze, sense-stench, sense-glitter** (o loro assenza)

Aggiorna la rappresentazione del mondo conosciuto

# Il cacciatore in Jess

## ▪ THINK

Rappresentazione interna: l'insieme dei fatti `cave` (vedere `deftemplate`)  
Posizione del nodo nel grafo + attributi (conoscenze del cacciatore)

Elaborazione dei dati percettivi

`evaluate-breeze`, `evaluate-stench`, `evaluate-glitter` (o loro assenza)  
Aggiorna la rappresentazione del mondo conosciuto, con estensione ai nodi *adiacenti*

Conclusioni, per inferenza (*beliefs*)

`safe-cave`, `safe-cave2`, `safe-cave3`

*Desideri*: lista delle intenzioni del cacciatore, con intensità (*strength*)

`desire-to-leave-caves`, Solo a scopo raggiunto

`add-desire-to-head-for-the-exit`

`lust-for-gold` Massima intensità

`add-desire-to-go-to-safe-adjacent-cave`, Maggiore intensità

`add-desire-to-go-to-safe-distant-cave`

`add-desire-to-go-to-risky-adjacent-cave`, Minore intensità

`add-desire-to-go-to-risky-distant-cave` (comprensibilmente ...)

# Il cacciatore in Jess

- PLAN

Rappresentazione interna: l'insieme dei fatti `cave` (vedere `deftemplate`)

Rappresentazione interna: l'insieme dei fatti `desire` (vedere `deftemplate`)

Una sola intenzione si trasforma in goal

**choose-desire**

Si sceglie l'intenzione a intensità maggiore

# Il cacciatore in Jess

- ACT

Rappresentazione interna: l'insieme dei fatti `cave` (vedere `deftemplate`)

Rappresentazione interna: l'insieme dei fatti `desire` (vedere `deftemplate`)

Rappresentazione interna: un solo fatto `goal` (vedere `deftemplate`)

Il `goal` si trasforma in azione

`pickup-gold`, `go-to-adjacent-cave`

Azioni dirette

`move-toward-distant-cave`

Azione immediata, verso un'obiettivo non immediato

# Il cacciatore in Jess

- Iterazione delle fasi

Transizione di stato

Iniziale

**genesis-to-simulate**

Main loop

**simulate-to-sense**

**sense-to-think**

**think-to-plan**

**plan-to-act**

**act-to-simulate**

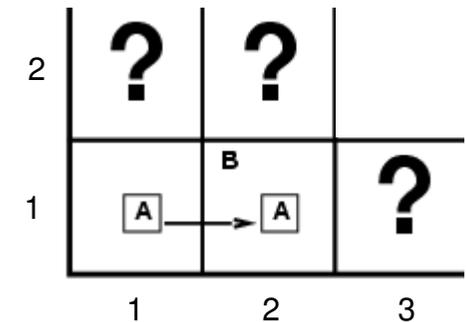
(Perchè hanno tutte priorità bassa?)

# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

## Puntate precedenti

GENESIS: Creazione del mondo  
 SIMULATE: Il cacciatore entra in (1,1)  
 SENSE: Non sente nulla  
 THINK: ...  
 PLAN: ...  
 ACT: Si muove in (2,1)  
 SIMULATE: ...  
 SENSE: Sente brezza



- Osservazioni (fatti)

$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$

- Capacità di ragionamento (regole + fatti di carattere generale)

$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$

# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

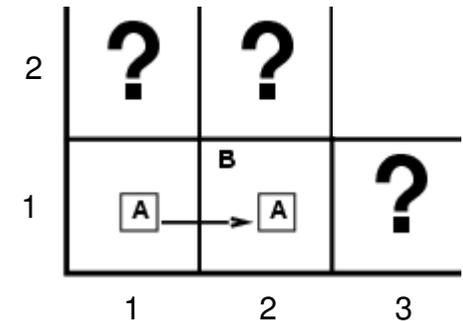
$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$

Osservazioni ( $\Sigma$ ):

$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$

KB è un insieme di fbf

$KB \equiv \Gamma \cup \Sigma$



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

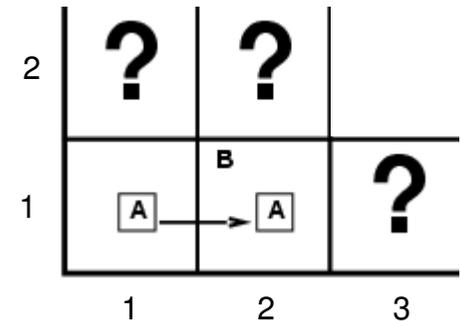
$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$

KB è un insieme di fbf

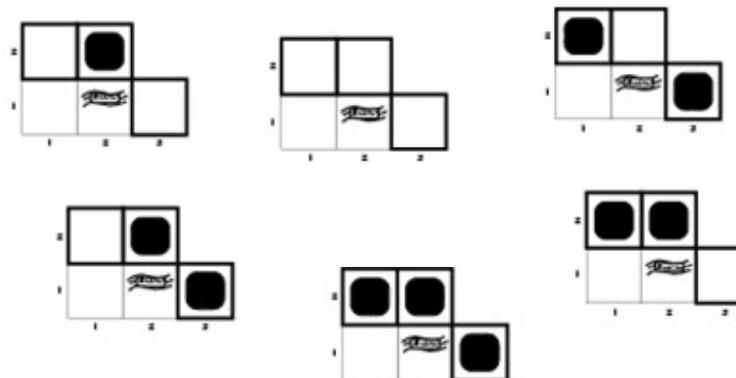
$$KB \equiv \Gamma \cup \Sigma$$

I modelli di  $\Sigma$

La descrizione dei mondi possibili compatibili con le osservazioni  $\Sigma$



Tutti i mondi possibili (nell'ipotesi *Safe or Pit*)



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

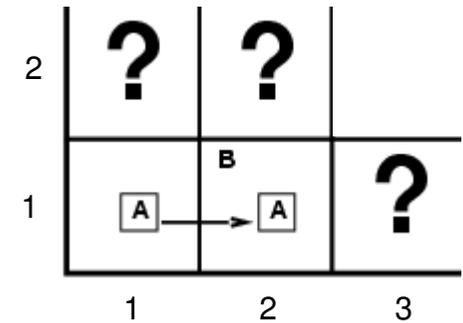
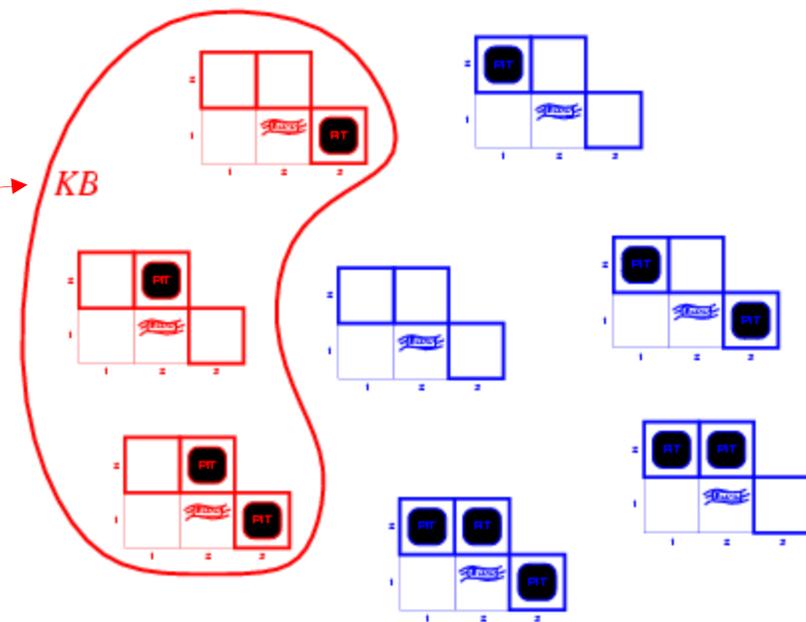
$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$

KB è un insieme di fbf

$$KB \equiv \Gamma \cup \Sigma$$

I modelli di KB

La descrizione dei  
mondi possibili  
compatibili con KB



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

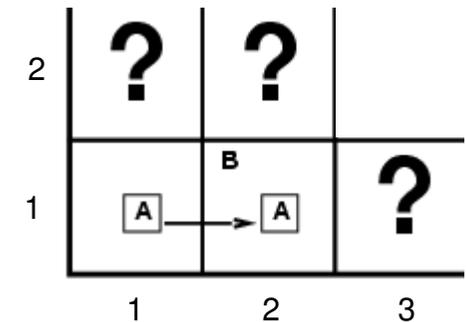
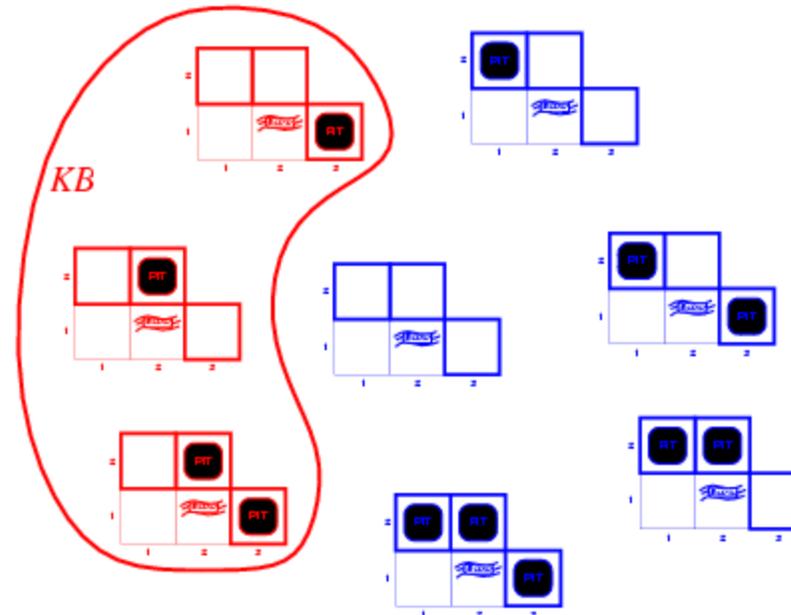
$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$

KB è un insieme di fbf

$$KB \equiv \Gamma \cup \Sigma$$

Le conseguenze logiche di KB

Sono le fbf i cui modelli  
includono quelli di KB



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$

KB è un insieme di fbf

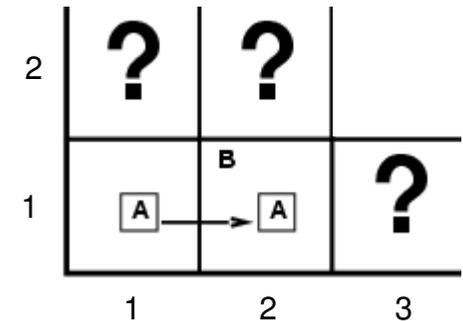
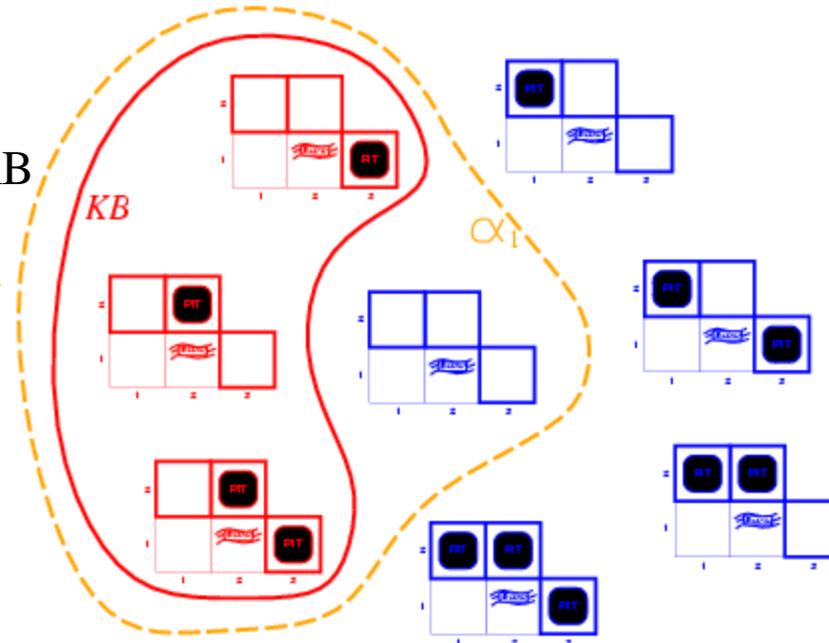
$$KB \equiv \Gamma \cup \Sigma$$

Le conseguenze logiche di KB

Sono le fbf i cui modelli includono quelli di KB

Esempio 1:

$$KB \models Safe(1, 2)$$



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

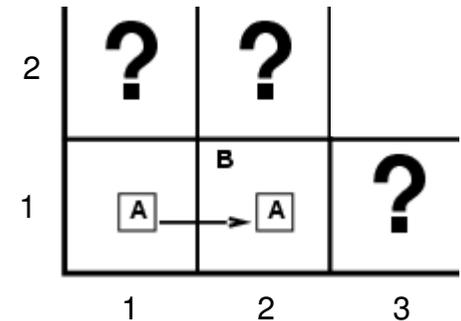
Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$



KB è un insieme di fbf

$$KB \equiv \Gamma \cup \Sigma$$

Le conseguenze logiche di KB

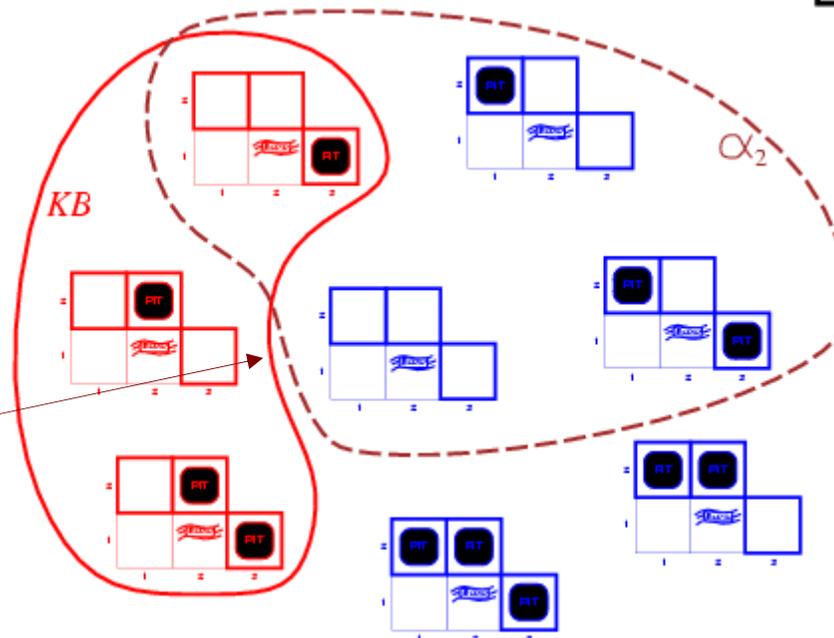
Sono le fbf i cui modelli includono quelli di KB

Esempio 1:

$$KB \models Safe(1, 2)$$

Esempio 2:

$$KB \not\models Safe(2, 2)$$



# Il cacciatore logico

- Base di conoscenza = capacità di ragionamento + osservazioni

Base di conoscenza (KB) attuale

Capacità di ragionamento o programma ( $\Gamma$ ):

$$\forall x_1 \forall y_1 \forall x_2 \forall y_2 ((\neg Breeze(x_1, y_1) \wedge Adjacent(x_1, y_1, x_2, y_2)) \rightarrow Safe(x_2, y_2))$$

Osservazioni ( $\Sigma$ ):

$$Safe(1,1), \neg Breeze(1,1), Safe(2,1), Breeze(2,1)$$

KB è un insieme di fbf

$$KB \equiv \Gamma \cup \Sigma$$

I teoremi di KB

Sono le fbf derivabili da KB

$$\{\varphi \mid KB \vdash \varphi\}$$

Se il metodo di calcolo è corretto e completo:

$$\{\varphi \mid KB \vdash \varphi\} \equiv \{\varphi \mid KB \models \varphi\}$$

