Inteligência artificial

Agentes lógicos

Adaptação dos slides de **Stuart Russel** e **Peter Norvig**, disponíveis em **aima.cs.berkeley.edu**

Estrutura da apresentação

- Sistemas Baseados em Conhecimento;
- Exemplo Pedagógico: *Mundo de Wumpus*;
- Lógica proposicional (booleana);
- Noções rudimentares sobre inferência.

Agentes baseados em conhecimento

- Base de conhecimento (BC) = conjunto de sentenças numa linguagem formal;
- Abordagem declarativa para construir um agente:
  - **Tell**: informar o que o agente precisa saber;
  - **Ask:** para o agente obter respostas, que *devem* seguir do que foi informado à BC;
- Agentes podem ser vistos no nível de conhecimento

  ex., o que eles sabem, não importando a implementação;
- Ou no nível de implementação
  - ex., estruturas de dados na BC e algoritmos que as manipulem;

Agentes baseados em conhecimento

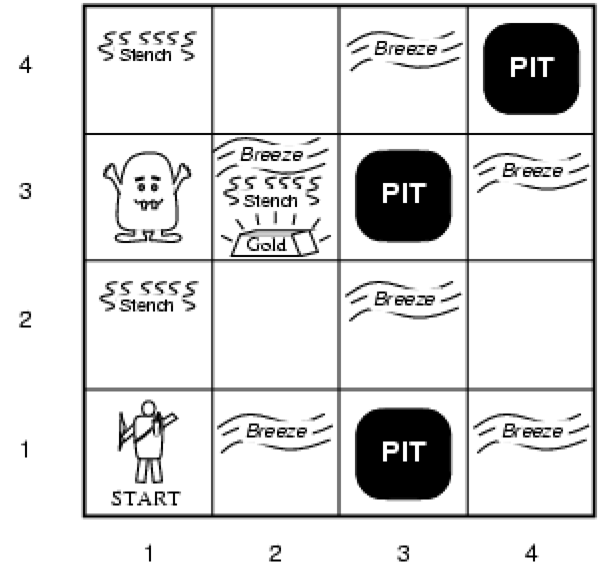
```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

- O agente precisa estar apto para:
  - Representar estados, ações etc;
  - Incorporar novas percepções;
  - Atualizar representações internas do mundo;
  - Deduzir propriedades ocultas do mundo;
  - Deduzir ações apropriadas;

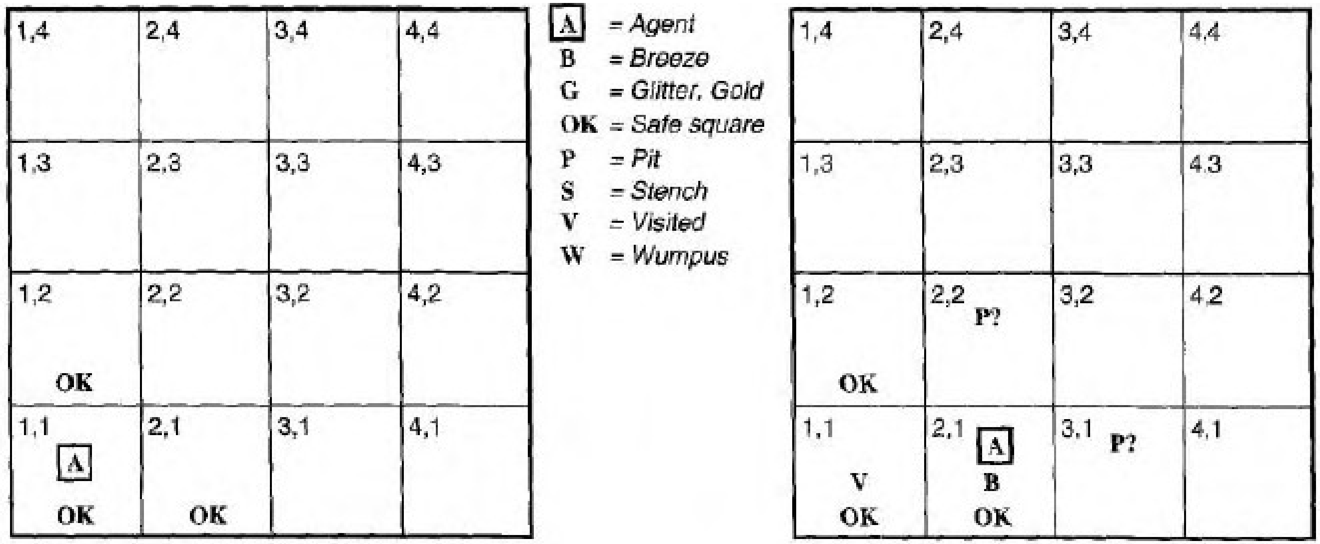
Exemplo: Mundo de Wumpus

- Posições adjacentes ao Wumpus são malcheirosas;
- Posições adjacentes aos buracos (*pit*) apresentam brisa;
- Posição onde está o ouro brilha;
- ....
- Sensores:
  - cheiro, brisa, brilho, grito do Wumpus etc.
- Atuadores:
  - direita, esquerda, para frente, para trás,
  - pegar, atirar etc.



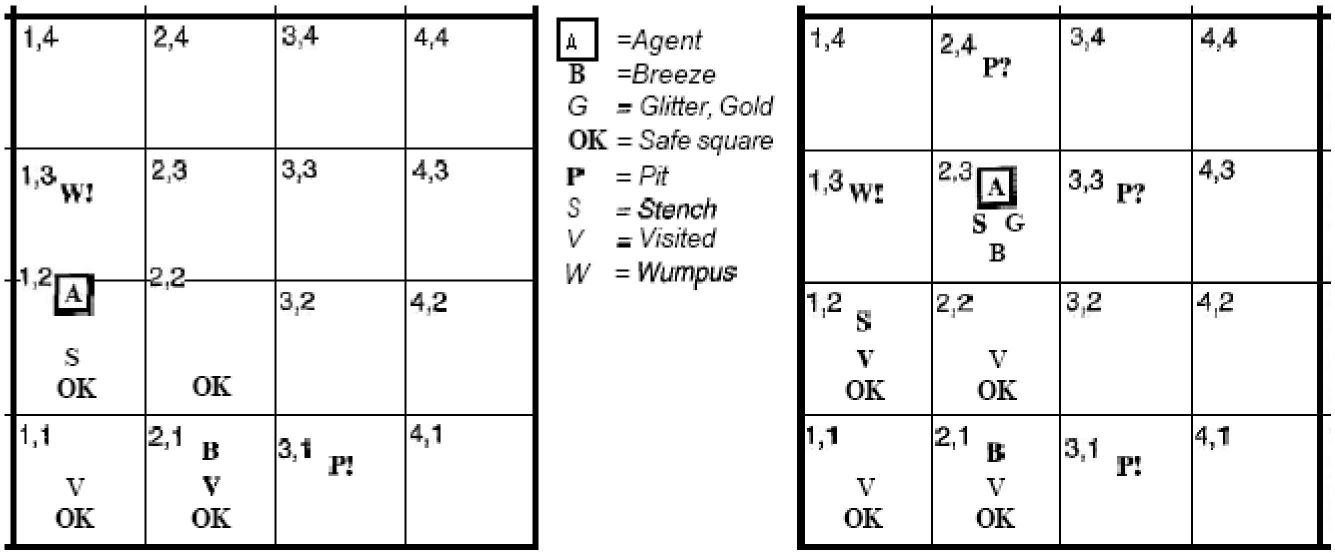
- O mundo de Wumpus é um ambiente totalmente observável, determinístico, episódico, estático e discreto.

Explorando o mundo de Wumpus



- [1,1]: Base de Conhecimento inicial do agente contém as regras do ambiente.
- [2,1]: Brisa (B – *breeze*) indica que existe um buraco (P – *pit*) em:
  - [2,2] ou [3,1].
- Retornar para [1,1] para tentar encontrar outra posição "segura".

Explorando o mundo de Wumpus



- Em [1,2] há mal-cheiro: Wumpus está em [1,3] ou [2,2].
  - Não pode estar em [2,2]; pois não há mal-cheiro em [2,1].
  - Logo, Wumpus está em [1,3].
  - Além disso [2,2] é seguro, pois não há brisa em [1,2].
  - Consequentemente, há buraco em [3,1].
- Mover para [2,2]; ….

Lógica e o mundo de Wumpus

- Sintaxe especifica como as sentenças devem ser formadas;
- Semântica define a verdade das sentenças em relação aos mundos possíveis;
- E.g., linguagem da aritmética:
  - $x+3 \geq y$ é uma sentença;
  - x2y > 2{} não é uma sentença;
  - $x+3 \geq y$ é verdadeira quando...

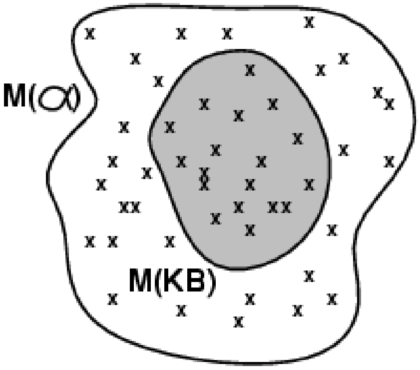
Noções sobre dedução

$BC \models \alpha$

- α *decorre logicamente* de BC.
- É possível deduzir logicamente a sentença α da base BC se, e somente se, α é verdadeiro em todos os mundos em que *BC* é verdadeiro.
- Ex., De $(x+y = 4)$ *decorre logicamente* de $(4 = x+y)$.

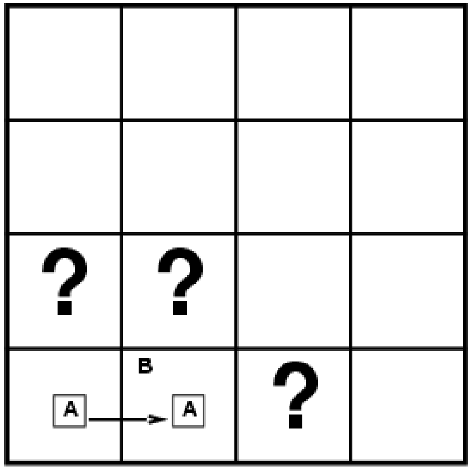
Noções sobre Modelos

- Os lógicos pensam em termos de *modelos*, que são mundos formalmente estruturados com respeito à "verdade" que será avaliada;
- *m* é um modelo de uma sentença se:
  - α é verdadeiro em *m*.
- *M(α)* é o conjunto de todos os modelos de α.
- BC ⊨ α se *M(BC)* ⊆ *M*(α).
- Em todos os modelos em que BC é verdadeiro:
  - α é verdadeiro.



Mundo de Wumpus

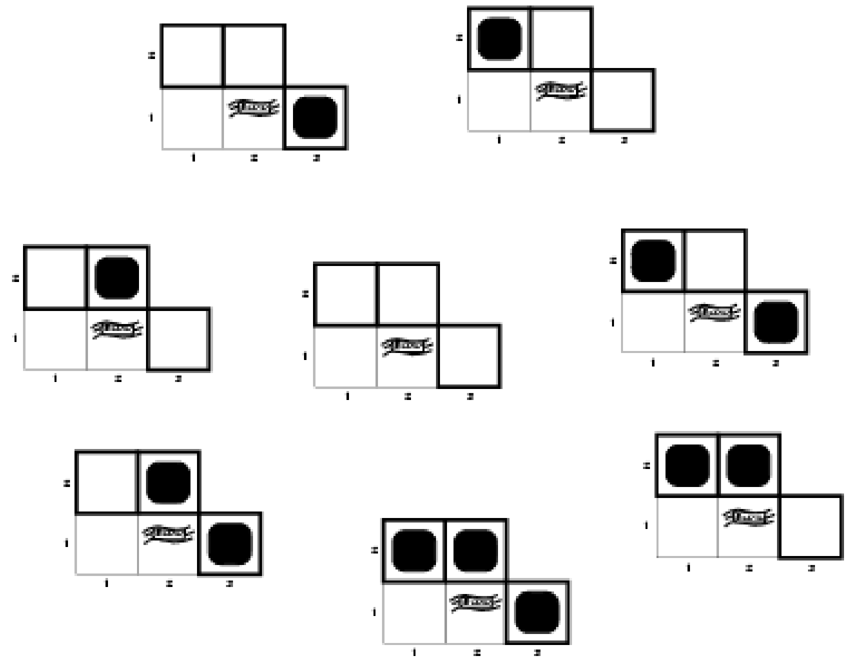
Situação após se mover de [1,1] para [2,1]:



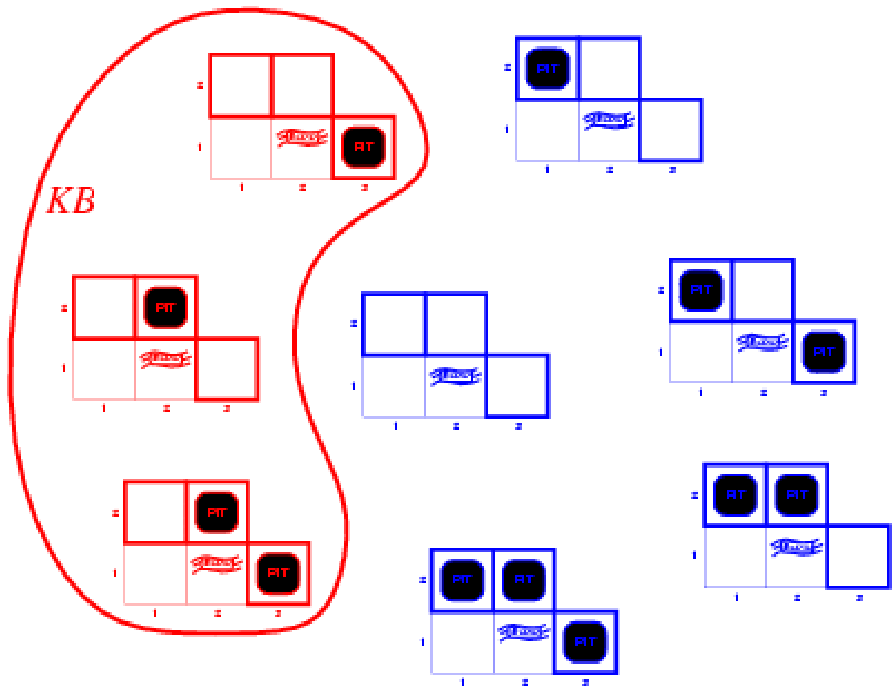
Modelos possíveis para BC relativa aos buracos (*pits*):

- 3 opções Booleanas - 8 modelos possíveis:

Modelos para Wumpus

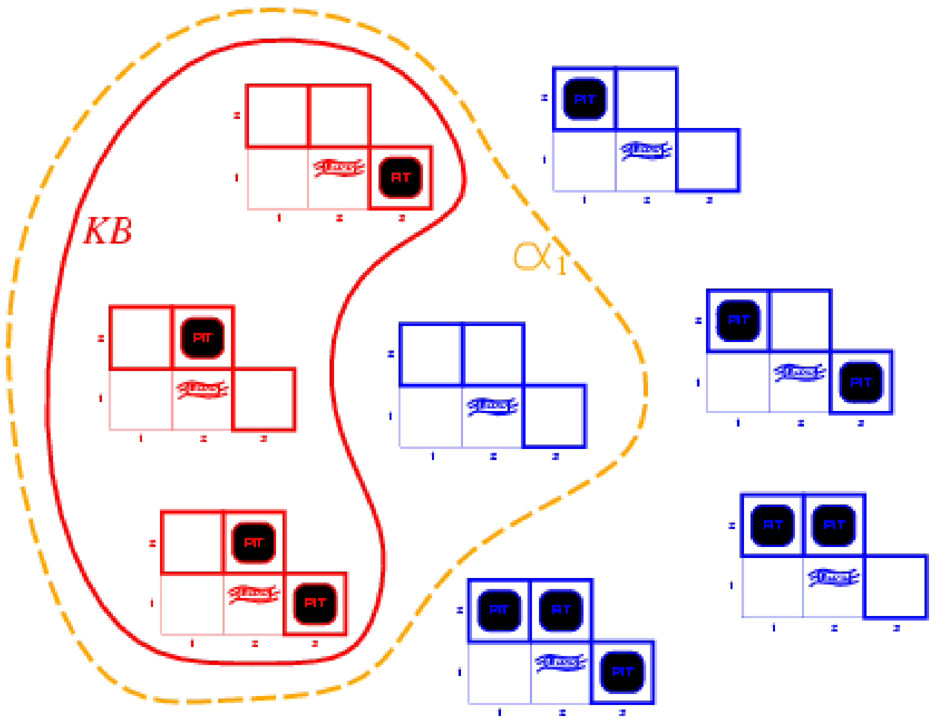


Modelos para Wumpus



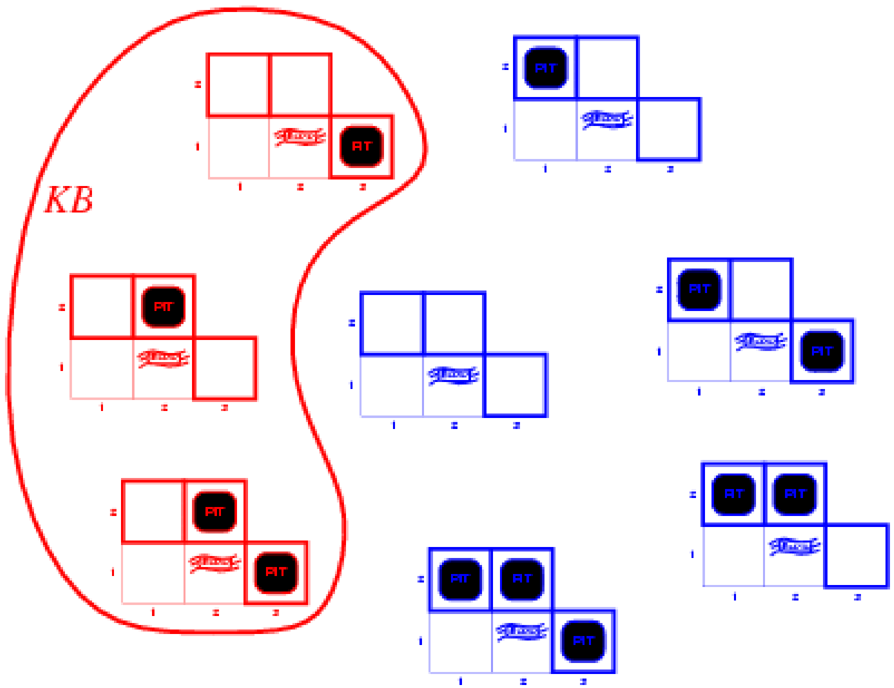
- BC = regras do mundo + observações.

Modelos para Wumpus

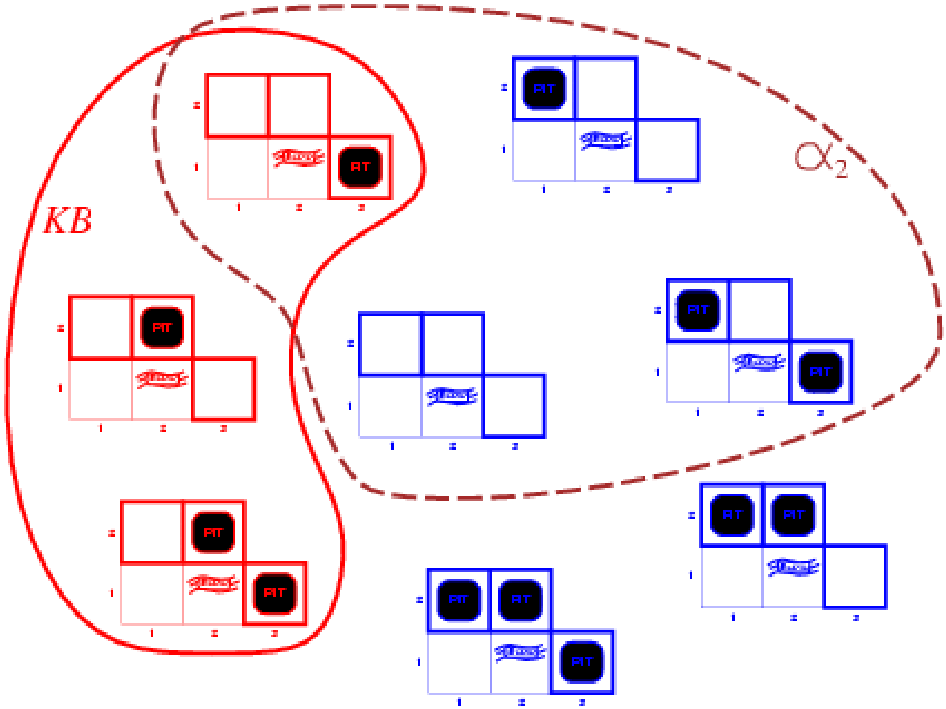


- α_1= "[1,2] é segura";
- *BC* ⊨ α_1, provada por verificação do modelo (*model checking*)

Modelos para Wumpus



Modelos para Wumpus



- α_2 = "[2,2] é seguro", $KB \not\models \alpha_2$

Noções sobre inferência

- $BC \vdash_i \alpha$: α é derivável de *BC* pelo algoritmo *i*.
- Consistência: *i* é robusto (preserva a verdade) se:
  - $BC \vdash_i \alpha$, então é verdade que $BC \models \alpha$.
- Completude: *i* é completo se:
  - $BC \models \alpha$, então é verdade que $BC \vdash_i \alpha$

Sistemas de lógica

- Um **sistema de lógica** geralmente consiste de:
  - *Sintaxe:* Símbolos e combinações de símbolos que constituem a linguagem lógica.
  - *Semântica:* Significado para as partes da linguagem, permitindo sua interpretação.
  - *Teoria da Prova*: Conjunto de regras de inferência.

Lógica/Cálculo proposicional

- **Sintaxe (Vocabulário):**
  - Constantes lógicas *True* e *False*;
  - Proposições simples P, Q, etc;
  - Conectivos (operadores) lógicos $\neg$, $\wedge$, $\vee$, $\Rightarrow$ e $\Leftrightarrow$;
  - Parêntesis ( ).

Sintaxe (gramática)

- Constantes lógicas *True* e *False*;
- Símbolos proposicionais (por exemplo, *P* ou *Q*);
- Conectivos lógicos de *Negação* ($\neg$), *Conjunção* ($\wedge$), *Disjunção* ($\vee$), *Condicional / Implicação* ($\Rightarrow$) e *Bicondicional / Equivalência* ($\Leftrightarrow$);
- O uso de parênteses ( ) para estabelecer precedência aos operadores.
- Prioridade dos conectivos
  - 1ª: $\neg$;
  - 2ª: $\wedge$, $\vee$;
  - 3ª: $\Rightarrow$;
  - 4ª: $\Leftrightarrow$;

Gramática Backus-Naur Form

$Sentença \rightarrow Sentença\_Atômica \mid Sentença\_Complexa$

$Sentença\_Atômica \rightarrow Verdadeiro \mid Falso \mid P \mid Q \mid \mathrm{R} \mid \ldots$

$Sentença\_Complexa \rightarrow (\ Sentença\ ) \mid \neg Sentença \mid Sentença\ Conectivo\ Sentença$

$Conectivo \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$

Tabela verdade para os conectivos lógicos

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Teoria da prova (regras de inferência)

- **1. Modus Ponens:** $$\left\{ \frac{\alpha, \alpha \Rightarrow \beta}{\beta} \right\}$$
- **2. Eliminação do E:** $$\left\{ \frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i} \right\}$$
- **3. Introdução do E:** $$\left\{ \frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n} \right\}$$
- **4. Introdução do OU:** $$\left\{ \frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n} \right\}$$

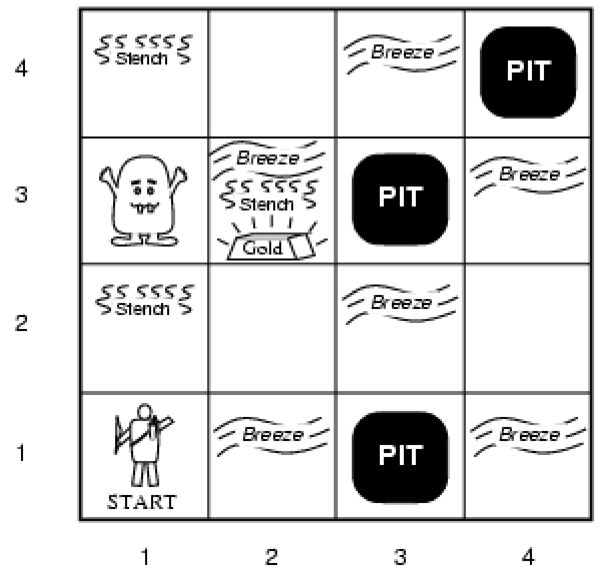
Teoria da prova (regras de inferência)

- **5. Eliminação da Dupla Negação:** $\left\{\dfrac{\neg\neg\alpha}{\alpha}\right\}$
- **6. Silogismo disjuntivo:**. $\left\{\dfrac{\alpha \vee \beta,\ \neg\beta}{\alpha}\right\}$
- **7. Resolução**: $\left\{\dfrac{\alpha \vee \beta,\ \neg\beta \vee \gamma}{\alpha \vee \gamma}\right\}$
- **8. Silogismo hipotético:** $\left\{\dfrac{\alpha \Rightarrow \beta,\ \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}\right\}$

Equivalência lógica

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

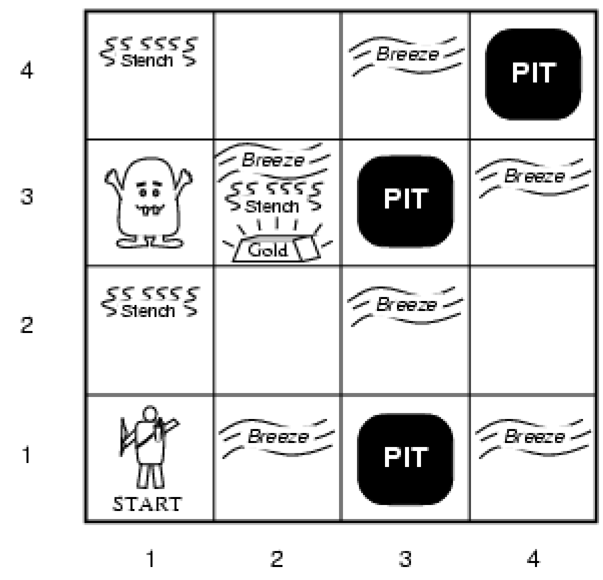
Mundo de Wumpus



- $P_{i,j}$ é verdadeiro se há um buraco em [i, j].
- $B_{i,j}$ é verdadeiro se há brisa em [i, j].
  - $\neg P_{1,1}$, $\neg B_{1,1}$, $B_{2,1}$

- Base de Conhecimento (BC):

- R_1: $\neg P_{1,1}$
- R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
- R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- ...
- Percepções de brisa do agente:
- R4: $\neg B_{1,1}$
- R5: $B_{2,1}$



BC: $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

- Objetivo da inferência lógica é decidir se BC $\models \alpha$ para alguma sentença α. Por exemplo:
  - BC $\models \neg P_{1,2}$?
  - BC $\models \neg P_{2,2}$?
- Algoritmo simples para inferência lógica:
  1. Enumerar todos os modelos possíveis;
  2. Verificar em quais desses modelos a BC é verdadeira;
  3. Verificar se é verdadeira em todo modelo em que BC é verdadeira.

Tabela verdade para a inferência

$\alpha_1 = \neg P_{1,2}$

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

Regras de inferência e equivalência

- Base de Conhecimento (BC):

  R_1: $\neg P_{1,1}$

  R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

  R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

  R_4: $\neg B_{1,1}$

  R_5: $B_{2,1}$

- Aplicando *equivalência (eliminação de bicondicional*) a R_2 obtemos:

  $R_6 : (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

Regras de inferência e equivalência ...

- Aplicando *Eliminação do E* em:

  R_6: $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$ temos:

  R_7: $(P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$
- Aplicando *Contraposição* em R_7 temos:

  R_8: $\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1})$
- Aplicando *Modus Ponens* em R_8 e considerando a percepção em R_4: $\neg B_{1,1}$:

  R_9: $\neg (P_{1,2} \vee P_{2,1})$
- Finalmente, aplicando regra de *De Morgan* em R_9:

  R_{10}: $\neg P_{1,2} \wedge \neg P_{2,1}$

Regras de inferência e equivalência ...

- O agente retorna de [2,1] e [1,1] e vai para um [1,2], onde percebe odor:

$R_{11:}\ \neg B_{1,2}$

R_{12}: $B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

- Pelo processo aplicado em $R_{10,}$ pode-se derivar a ausência de poços em [2,2] e [1,3]:

$R_{13} = \neg P_{2,2}$

$R_{14} = \neg P_{1,3}$

Regras de inferência e equivalência ...

- Eliminação de bicondicional em R_3 seguida por *modus ponens* com R_5

  R_{15}: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$
- Não há buraco em [2,2], $\neg P_{2,2}$

  R_{16}: $P_{1,1} \vee P_{3,1}$
- Não há buraco em [1,1], $\neg P_{1,1}$:

  $R_{17:}$ $P_{3,1}$

Métodos de prova

- Dois tipos:
  - Aplicação de Regras de Inferência
  - Verificação de Modelos
- *Buscar por Provas* não é mais eficiente do que enumerar modelos.
- *Monotonicidade*: o conjunto de consequências lógicas só pode aumentar à medida que informações são adicionadas à BC:
  - Se $BC \models \alpha$ então $BC \wedge \beta \models \alpha$

Referências

- Stuart Russel e Peter Norvig, **Inteligência Artificial**, 2ª edição, Editora Campus, 2004.