Conhecimento e Raciocínio
Lógica Proposicional

Agente Baseado em Conhecimento ou Sistema Baseado em Conhecimento

- Representa conhecimento sobre o mundo em uma ***linguagem formal*** (*KB*)

- Raciocina sobre o mundo fazendo ***inferências*** na linguagem de representação adotada na KB

$$KB \vdash_i \alpha$$

- Decidi que ação tomar ***inferindo*** sobre qual ação selecionada é a melhor

Inferência Lógica

- *raciocínio ou inferência*: processos para se chegar a uma conclusão (exemplo no mundo de wumpus: posição (2,2) é segura?)
- *inferência lógica*: processo que implementa a relação de encadeamente entre sentenças
- existem várias maneira de se implementar uma inferência lógica

Modelos

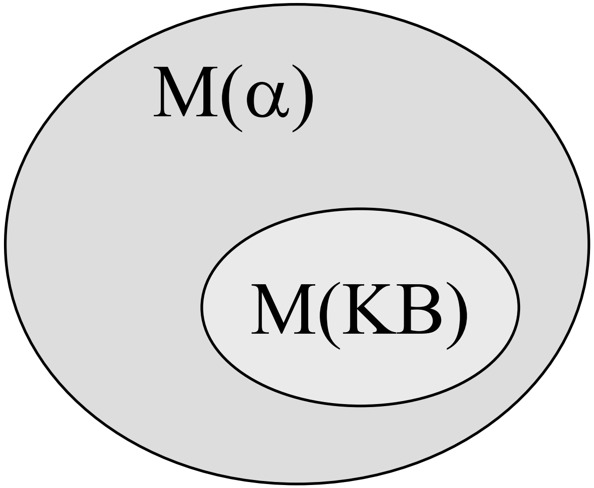
Lógicos pensam em termos de modelos, que são mundos formalmente estruturados com relação à verdade das sentenças

Dizemos que *M* é um modelo de uma sentença α se α é verdade em *M*

Seja $M(\alpha)$ o conjunto de todos os modelos de α

Então $KB \models \alpha$ sse $M(KB) \subseteq M(\alpha)$

interpretações	A	B	α: A=>B
I_1	F	F	V
I_2	F	V	V
I_3	V	F	F
I_4	V	V	V



Sentenças Válidas e Satisfatíveis

- Uma sentença é valida (ou necessariamente verdadeira) sse ela for verdade em todas as possíveis interpretações (coluna toda V)

Ex.: $A \vee \neg A$, $A \rightarrow A$, $(A \wedge (A \rightarrow B)) \rightarrow B$

$KB \models S$ sse $(KB \rightarrow S)$ é válida (*tautologia*) *independe do significado*

- Uma sentença é <u>*satisfatível*</u> se ela for verdade em <u>algum</u> modelo
- Uma sentença é <u>*não-satisfatível*</u> se, em <u>nenhum</u> modelo, ela é verdade. Ex: $A \wedge \neg A$

$KB \models S$ sse $(KB \wedge \neg S)$ é <u>*não-satisfatível*</u>

Satisfatibilidade - exemplo

- *Uma sentença é satisfatível sse for verdadeira para alguma interpretação em algum mundo*
- "Existe um abismo em [2,2]" pode ser verdadeira para algum mundo do Wumpus
- *Uma sentença é não-satisfatível sse não existir nenhuma interpretação em algum mundo em que ela seja verdadeira*
- "Não existe um abismo em [2,2] e existe um abismo em [2,2]"

$$\neg A \wedge A$$

Verificação de sentenças válidas

- A pergunta "[2,2] é OK?" deve ser feita da seguinte forma:

  "se KB é verdade então [2,2] é uma posição segura" é uma sentença válida?

- *KB* é uma conjunção de milhares de sentenças e *P* é uma sentença complexa

como construir um procedimento formal para verificação de sentenças válidas?

Equivalências lógicas e tautologias

- Duas sentenças α e β são logicamente equivalentes se são verdadeiras para o mesmo conjunto de modelos

$$\alpha \equiv \beta$$

Uma definição:

$$\alpha \equiv \beta \quad \text{se e somente se} \quad \alpha \models \beta \text{ e } \beta \models \alpha$$

- Teorema da dedução:
- $\alpha \models \beta$ se e somente se a sentença $\alpha \rightarrow \beta$ é válida

Equivalências lógicas

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	comutativa de $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	comutativa de $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\gamma \wedge (\beta \wedge \alpha))$	associativa de $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\gamma \vee (\beta \vee \alpha))$	associativa de $\vee$
$(\alpha \Leftrightarrow \beta) \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$	eliminação da bicondicional
$\neg(\neg\alpha) \equiv \alpha$	eliminação da negação dupla
$(\alpha \rightarrow \beta) \equiv (\neg \beta \rightarrow \neg \alpha)$	contraposição
$(\alpha \rightarrow \beta) \equiv \neg\alpha \vee \beta$	eliminação da implicação
$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$	de Morgan
$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$	de Morgan

Lógica Proposicional

- sintaxe
- semântica
- validade e inferência
- modelos
- regras de inferência para LP
- provadores de teoremas: método de prova (e resolução de problema) de propósito geral

Semântica do Cálculo Proposicional

- Significado de uma sentença através da noção de <u>interpretações de sentenças</u>
  - atribuição de valores verdade para cada um dos símbolos e constantes proposicionais e
  - especificação do significado dos conectivos lógicos (tabelas verdade).
- A verdade da fórmula geral é calculada baseada nas leis da Lógica Booleana: como computar a verdade de sentenças envolvendo conectivos ($\wedge, \vee, \rightarrow, \leftrightarrow$) dados os valores verdade de seus constituintes ?

LP - Validade e inferência

- Validade e inferência
  - dada uma sentença podemos fazer uma tabela verdade com uma linha para cada uma das possíveis combinações de valores verdade para os símbolos proposicionais da sentença
  - para cada linha podemos calcular o valor da sentença inteira. Se a sentença for verdade para todas as linhas então a sentença é válida. Ex:

    $((P \vee H) \wedge \neg H) \rightarrow P$ é válida e portanto,

    se é verdade $((P \vee H) \wedge \neg H)$ podemos inferir P

Inferência Proposicional: Método da Enumeração

Seja α: AvB e KB: (AvC)^(Bv~C)

KB |= S ??

Verificação de todos os modelos possíveis de S: devem ser verdadeiros sempre que KB é verdade

•2^n mundos ou interpretações

A	B	C	AvC	Bv~C	KB	α
F	F	F				
F	F	V				
F	V	F				
F	V	V				
V	F	F				
V	F	V				
V	V	F				
V	V	V				

Inferência Proposicional: solução

A	B	C	$A \vee C$	$B \vee \neg C$	KB	S	$KB \rightarrow S$
F	F	F	F	V	F	F	V
F	F	V	V	F	F	F	V
F	V	F	F	V	F	V	V
F	V	V	V	V	V	V	V
V	F	F	V	V	V	V	V
V	F	V	V	F	F	V	V
V	V	F	V	V	V	V	V
V	V	V	V	V	V	V	V

$\boxed{KB \models S}$

LP - problema

Dada uma fórmula contendo n átomos, existem 2^n interpretações possíveis (atribuição de valores V/F aos átomos individuais da fórmula)

Regras de Inferência

- Tabelas verdade podem ser extendidas para classes de inferências
- Existem certos padrões de inferências que ocorrem com bastante frequência: *regras de inferência*
- podem ser usadas para fazer inferência sem termos que construir tabelas de verdade

Notações usadas para regras de inferência

- Sendo α , β sentenças da lógica proposicional (separadas por “,”)
- $\alpha \vdash \beta$, significa que β pode ser derivada por α através de inferências.
- Ou:

$$\frac{\alpha}{\beta}$$

para enfatizar que a regra de inferência não se trata de uma sentença da linguagem (isto é, de uma meta-linguagem)

Notações usadas para regras de inferência

$$\frac{\alpha}{\beta}$$

Sempre que algo na KB casar com o *padrão* acima da linha, a regra de inferência conclue o que está abaixo da linha

7 regras de inferência para a LP

$\dfrac{\alpha \Rightarrow \beta, \ \alpha}{\beta}$	Modus Ponens	Da implicaçăo e da premissa infere-se a conclusăo
$\dfrac{\alpha_1 \wedge \alpha_2 \wedge \ \dots \ \wedge \alpha_n}{\alpha_n}$	Eliminaçăo	Da conjunçăo infere-se qualquer α_n
$\dfrac{\alpha_1, \alpha_2, \ \dots, \ \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ \dots \ \wedge \alpha_n}$	Introduçăo da conjunçăo	De uma lista de sentenças infere-se a sua conjunçăo
$\dfrac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ \dots \ \vee \alpha_n}$	Introduçăo da disjunçăo	De uma sentença, infere-se sua disjunçăo com qualquer outra
$\dfrac{\neg \neg \alpha}{\alpha}$	Negaçăo dupla	De uma negaçăo dupla infere-se uma senetnça positiva
$\dfrac{\alpha \vee \beta, \neg \beta}{\alpha}$	Resoluçăo simples	Se uma das disjunçőes for falsa, pode-se inferir que a outra é verdade
$\dfrac{\alpha \vee \beta, \neg \beta \vee \gamma}{\alpha \vee \gamma}$	Resoluçăo	β , năo pode ser Verdade e Falso ao mesmo tempo

Métodos de prova

Podem ser divididos em dois tipos:

Verificação de modelos:

– enumeração na tabela verdade (correto e completo para LP, porém = 2^n para n símbolos proposicionais)

– busca heurística no espaço de modelos (correto mas não completo)

Aplicação de regras de inferência:

– garante a geração de novas sentenças a partir de velhas

– prova = uma sequência de aplicações de regras de inferência

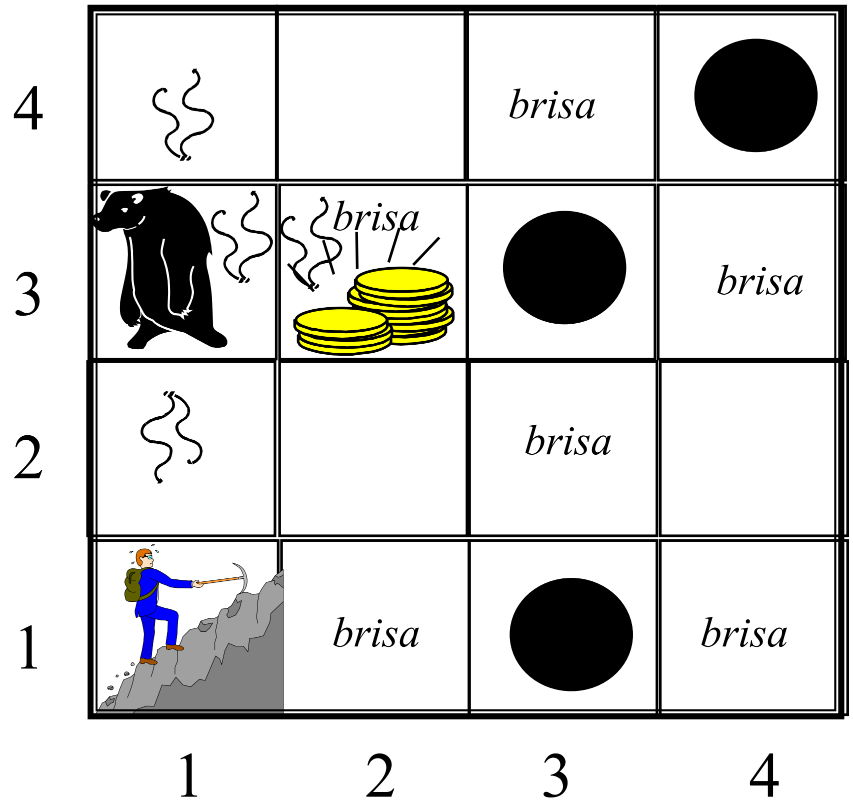
– pode-se usar regras de inferência como operadores em algoritmos de busca

Complexidade da prova lógica

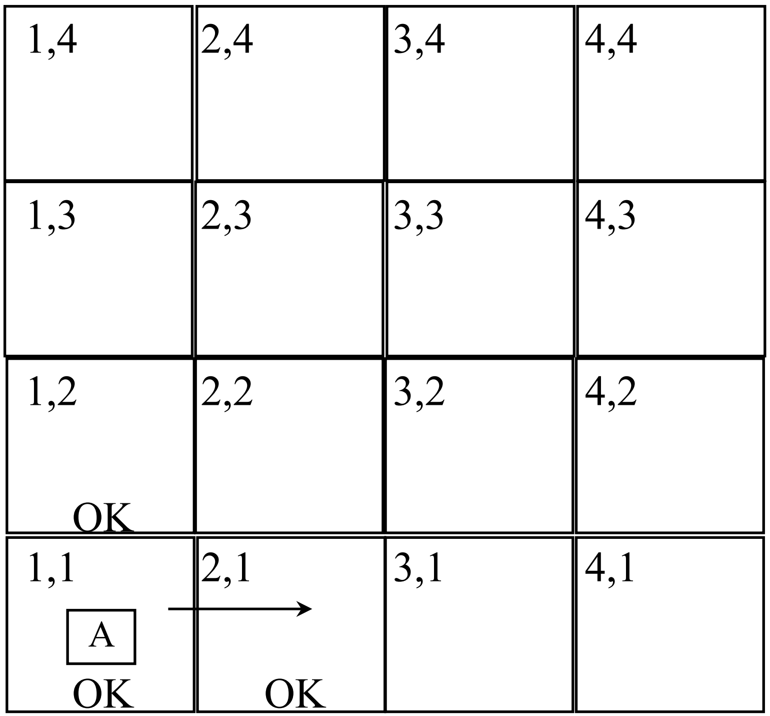
- **Monotonocidade:** uso local de regras de Inferência (se as sentenças derivadas de uma KB, continuam sendo derivadas da KB quando adicionamos novas sentenças a ela)
- **Sentenças de Horn**:
  - $P1 \wedge P2 \wedge \ldots \wedge Pn \Rightarrow Q$

  reduzem a complexidade para polinomial
  - Permitem que se use somente Modus Ponens
  - Importante: nem toda KB pode ser escrita nessa forma

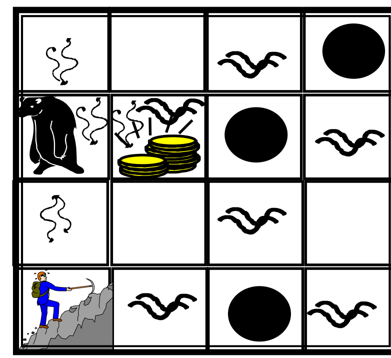
Um agente lógico no mundo de Wumpus



O mundo de Wumpus - LP

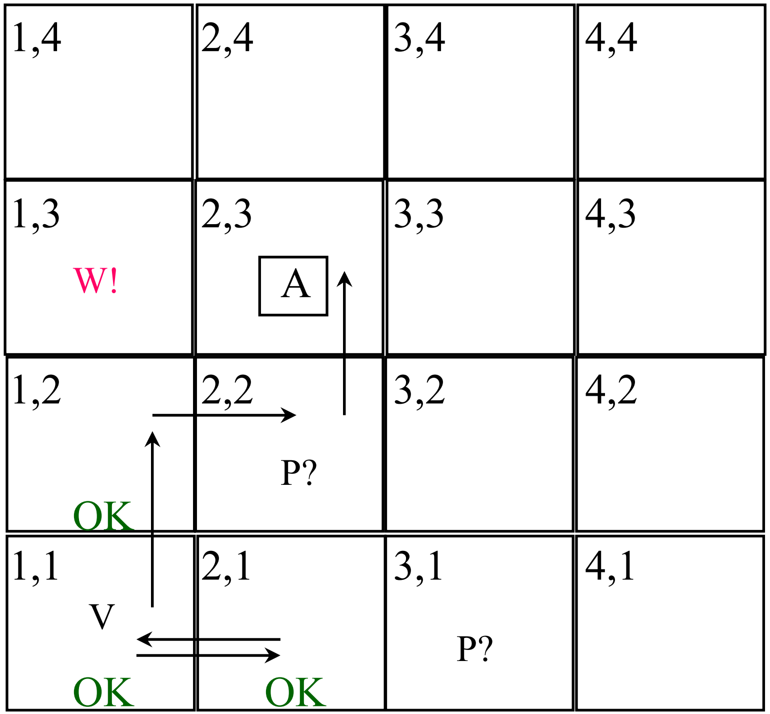


A = Agente
B = Brisa
G = Gold, brilho
OK = quadrado seguro
M = abismo
C = Cheiro
V = Visitado
W = Wumpus

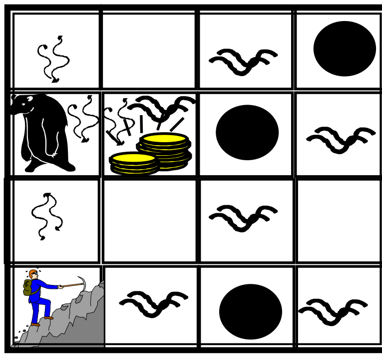


Percepção (i,j): [*<Cheiro>, <Brisa>, <Brilho>, <Choque>, <Grito>*]

O mundo de Wumpus - LP



A = Agente
B = Brisa
G = Gold, brilho
OK = quadrado seguro
M = abismo
C = Cheiro
V = Visitado
W = Wumpus



Percepção (1,1): [nada, nada, nada, nada, nada]
Percepção (2,1): [nada, *Brisa*, nada, nada, nada]
Percepção (1,2): [*Cheiro*, nada, nada, nada, nada]
Percepção (2,3): [*Cheiro*, *Brisa*, *Brilho*, nada, nada]

Situações de perigo

- Brisa em (1,2) e (2,1) => não existe ação segura
- Cheiro em (1,1) => não existe ação seguro

Uma Base de Conhecimento Simples

- Escolha de símbolos proposicionais
  - Existe um cheiro em (i,j)
  - Existe Wumpus em (i,j)
  - Existe brisa em (i,j) …
- Sentenças:
  - *Um quadrado tem um poço se um quadrado vizinho tem uma brisa*
  - *Um quadrado tem uma brisa se um quadrado vizinho tem um poço*
  - …

Uma Base de Conhecimento Simples

Escolha de símbolos proposicionais para as percepções:

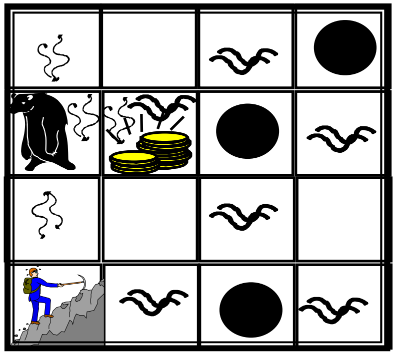
$S_{1,2}$ significa “Existe um cheiro em (1,2)”

$B_{2,1}$ significa “Existe uma brisa em (2,1)”

A sequência de percepções deve ser convertida em sentenças introduzidas na KB + sentenças válidas derivadas dessas sentenças de percepção

KB depois do terceiro movimento:

$\neg S_{1,1}$	$\neg B_{1,1}$
$\neg S_{2,1}$	$B_{2,1}$
$S_{1,2}$	$\neg B_{1,2}$

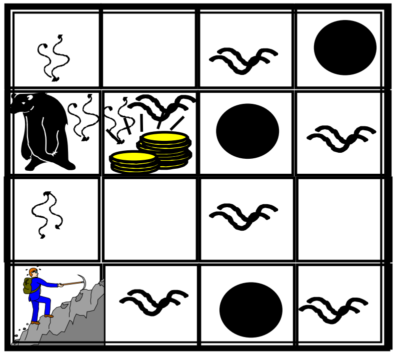


Uma Base de Conhecimento Simples

Escolha de símbolos proposicionais para a presença de poços e Wumpus:

$P_{1,2}$ significa "Existe um poço em (1,2)"

$W_{3,3}$ significa "O Wumpus mora em (3,3)"



Uma base de conhecimento simples

- *“Um quadrado tem um poço se um quadrado vizinho tem uma brisa”*

Uma base de conhecimento simples

- *"Um quadrado tem um poço se um quadrado vizinho tem uma brisa"*

$$B_{1,1} \rightarrow (P_{1,1} \lor P_{1,1})$$

Uma base de conhecimento simples

- *"Um quadrado tem um poço se um quadrado vizinho tem uma brisa"*

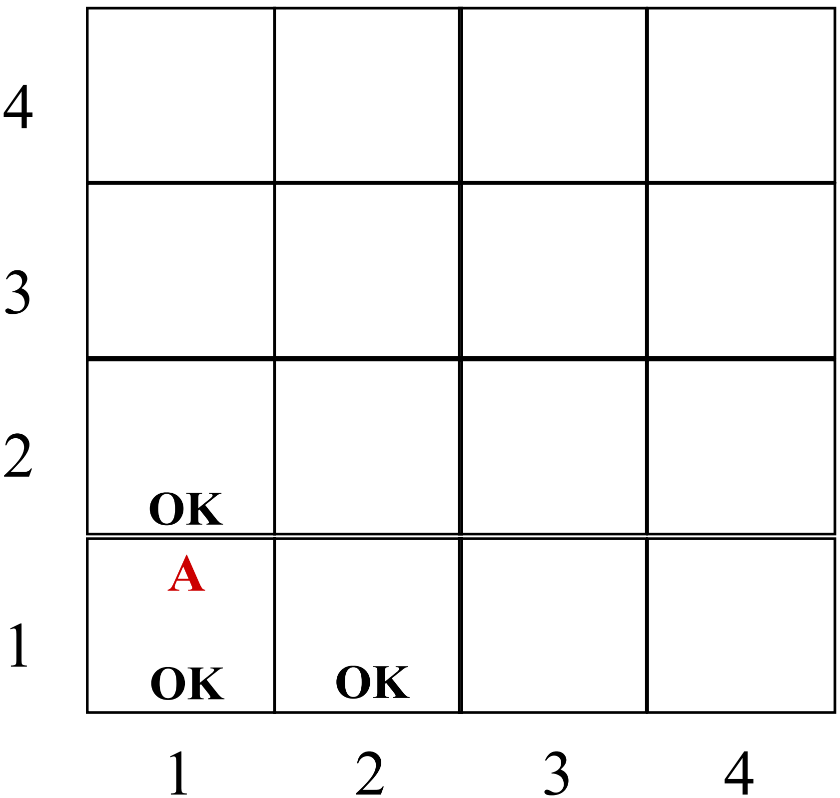
$$B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})$$

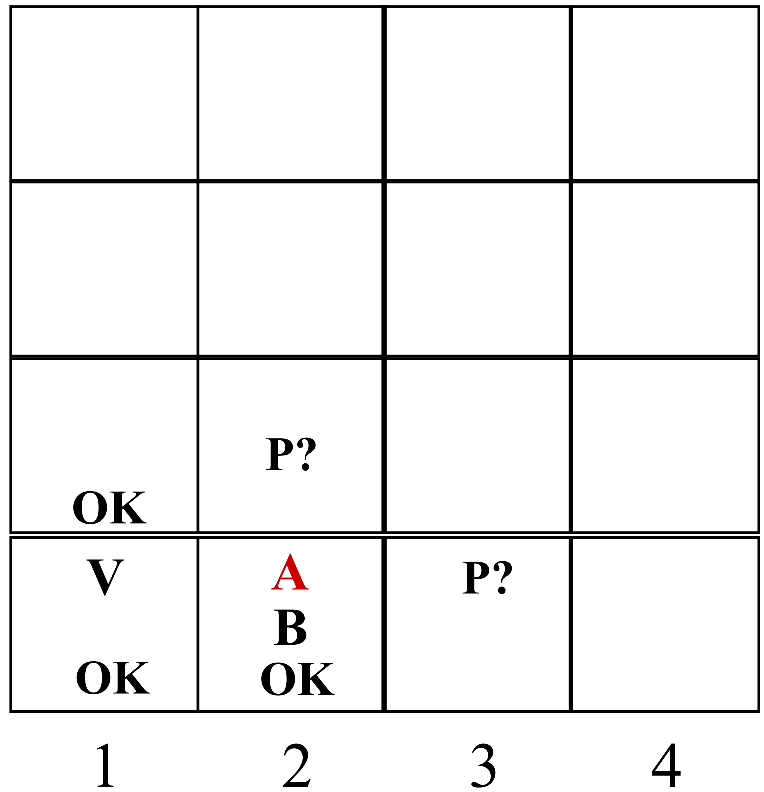
- É verdadeira no Mundo do Wumpus mas não elimina modelos em que $B_{1,1}$ é falsa e $P_{1,2}$ é verdadeira, violando assim as regras do jogo

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

Agente Lógico no Mundo do Wumpus

Percepção (i,j) ≡ [*<Cheiro>,<Brisa>,<Brilho>,<Choque,<Grito>*]





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

Uma base de conhecimento simples

R1: $\neg\ P_{1,1}$.

R2: $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$.

R3: $B_{,2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$.

R4: $\neg\ B_{1,1}$.

R5: $B_{2,1}$.

Que pode ser considerada uma única sentença: $R1 \land R2 \land R3 \land R4 \land R5$

Queremos provar que $\neg P_{1,2}$ é verdadeira

Eliminação da bicondicional em R2

R6: $(B_{1,1} \rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \rightarrow B_{1,1})$.

Eliminação de $\land$

R7: $((P_{1,2} \lor P_{2,1}) \rightarrow B_{1,1})$.

Equivalência lógica para contraposição

R8: $(\neg B_{1,1} \rightarrow \neg(P_{1,2} \lor P_{2,1}))$

Modus Ponens com R8 e R4

R9: $\neg(P_{1,2} \lor P_{2,1})$

Regra de De Morgan

R10: $\neg P_{1,2} \land \neg P_{2,1}$

A Base de Conhecimento

- Conhecimento sobre o ambiente. Por ex.:
  - Se não tem cheiro em um quadrado, o agente precisa saber que o Wumpus não mora no quadrado e em nenhum dos quadrados vizinhos (para todos os quadrados)

$R_{11}: \neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

$R_{12}: \neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

$R_{13}: \neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$

Se tem cheiro em (1,2) então o Wumpus deve morar em (1,2) ou em algum dos quadrados vizinhos

$R_{14}: S_{1,2} \Rightarrow W_{1,3} \wedge W_{1,2} \wedge W_{2,2} \wedge W_{1,1}$

Procurando o Wumpus

- queremos concluir $W_{1,3}$
- $KB \rightarrow W_{1,3}$ é válida?
  - Considerando 12 símbolos proposicionais:

    $S_{1,1}$, $S_{2,1}$, $S_{1,2}$,$W_{1,1}$,$W_{1,2}$,$W_{2,1}$,$W_{2,2}$,$W_{3,1}$, $W_{1,3}$, $B_{1,1}$, $B_{2,1}$, $B_{1,2}$

    e aplicando o método da Tabela Verdade teriamos: $2^{12} = 4096$ linhas e toda linha que KB é verdade, $W_{1,3}$ também deve ser verdade.
- Ao invés disso usamos as regras de inferência

Procurando o Wumpus (pag. 176)

1. Modus ponens:

$$\frac{R_1: \neg \mathbf{S_{1,1}} \Rightarrow \neg \mathbf{W_{1,1}} \wedge \neg \mathbf{W_{1,2}} \wedge \neg \mathbf{W_{2,1}} \quad , \neg \mathbf{S_{1,1}}}{\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}}$$

2. Eliminação (*And-Elimination*) no resultado acima:

$$\frac{\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}}{\neg W_{1,1} , \neg W_{1,2} , \neg W_{2,1}}$$

… continuação da prova em 7 passos (pag 175, AIMA)

Selecionando ações

- Como relacionar estado do mundo e ação?
- Exemplo de regra para o caso em que o agente está em (1,1) de frente para o leste:

$$A_{1,1} \wedge Leste_{\,a} \wedge W_{2,1} \rightarrow \neg \text{ ação-parafrente}$$

Lógica Proposicional é fraca

- só para a completar a regra anterior: ”não vá para frente se o Wumpus estiver na sua frente” seriam necessárias 64 regras (16 posições x 4 orientações do agente)
- Mudanças no tempo:

$$A^0_{1,1} \wedge Leste^0_a \wedge W^0_{2,1} \rightarrow \neg \text{ação-parafrente}^0$$

- Para 100 tempos: 6400 regras proposicionais (LPO reduz para 1 regra !!! Introdução de objetos e relações)

Prova como algoritmo de busca

Aplicação de regras de inferência:

- prova = uma sequência de aplicações de regras de inferência
- pode-se usar regras de inferência como operadores em algoritmos de busca:
  - para frente a partir da base de conhecimento inicial, aplicando as regras de inferência, equivalências lógicas e tautologias para derivar a sentença que se deseja provar ou
  - Para trás a partir da sentença que se deseja provar até derivar as sentenças da base de conhecimento

Prova como algoritmo de busca

- Lógica proposicional é NP-completa ➔ no pior caso a busca de provas não é mais eficiente que a enumeração de modelos
- Porém, em muitos casos práticos, encontrar uma prova pode ser muito mais eficiente porque podemos ignorar proposições irrelevantes

Prova por resolução

- Procedimento de inferência completo para toda a lógica proposicional
- Transformar a KB num conjunto de **cláusulas** (disjunção de literais)
- Teorema da completeza da resolução em lógica proposicional (teorema básico da resolução)
  – *Se um conjunto de cláusulas é não-satisfatível então o fechamento de resolução dessas cláusulas contém a cláusula vazia*