Sistemas Baseados em Conhecimento

Profa. Dra. Sarajane Marques Peres

junho de 2018

Disciplina: Inteligência Artificial
Bacharelado em Sistemas de Informação
http://www.each.usp.br/si

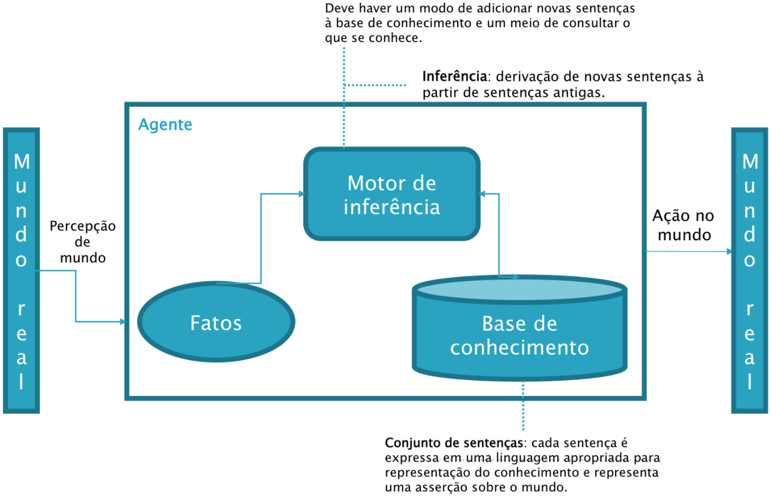
Sistemas Baseados em Conhecimento ou ...

- Sistemas Especialistas
- Agentes Baseados em Conhecimento
- Agentes Lógicos

Material desenvolvido baseado em:

- Russel, S.; Norvig, P. Inteligência Artificial. 2a. edição. Editora Campus, 2004. Capítulos: 7

Sistema (agente) baseado em conhecimento



Implementação de um agente baseado em conhecimento

- **função** AGENTE-BC(percepção) **retorna** uma *ação*
    - variáveis: BC - base de conhecimento; t - contador de tempo;
    - Looping
        - TELL(BC, CRIAR-SENTENÇA-DE-PERCEPÇÃO (percepção, t)
        - *ação* $\leftarrow$ ASK(BC, CRIAR-CONSULTA-AÇÃO(t))
        - TELL(BC, CRIAR-SENTENÇA-DE-AÇÃO(*ação*, t)
        - $t \leftarrow t + 1$
    - **retorna** *ação*

- TELL: informa à base de conhecimento o que o agente percebeu.
- CRIAR-SENTENÇA-DE-PERCEPÇÃO: recebe uma percepção e um instante de tempo e retorna uma sentença afirmando que o agente percebeu a percepção no instante informado.
- ASK: o agente pergunta à base de conhecimento o que ele deve executar.
- CRIAR-CONSULTA-AÇÃO: toma um instante de tempo como entrada e retorna uma sentença que pergunta que ação deve ser executada nesse instante.
- CRIAR-SENTENÇA-AÇÃO: informa à base de conhecimento que a ação foi executada.

Implementação de um agente baseado em conhecimento

- **função** AGENTE-BC(percepção) **retorna** uma *ação*
    - variáveis: BC - base de conhecimento; t - contador de tempo;
    - Looping
        - TELL(BC, CRIAR-SENTENÇA-DE-PERCEPÇÃO (percepção, t)
        - *ação* $\leftarrow$ ASK(BC, CRIAR-CONSULTA-AÇÃO(t))
        - TELL(BC, CRIAR-SENTENÇA-DE-AÇÃO(*ação*, t)
        - $t \leftarrow t + 1$
    - **retorna** *ação*

TELL e ASK implementam o mecanismo de inferência que informa à base de conhecimento o que o agente percebe, pergunta à base de conhecimento que ação deve executar e informa à base de conhecimento qual ação foi executada. A resposta a essa consulta envolve a implementação de um processo de raciocínio sobre o mundo atual e sobre os resultados de sequências de ações possíveis. Então, o agente registra a escolha e executa a ação.

Sistema (agente) baseado em conhecimento

Abordagem declarativa

O agente baseado em conhecimento usa uma descrição no **nível de conhecimento**. Para construir essa descrição os projetistas especificam apenas o que o agente sabe e quais são suas metas. O agente se adapta a esse cenário e corrige seu comportamento.

Exemplo

Um táxi automatizado tem a meta de entregar um passageiro em Marin County. Considere que ele sabe que está em San Fransciso e que a ponte Golden Gate é a única ligação entre os dois locais. Então, é esperado que ele cruze a ponte Golden Gate porque ele sabe que isso o levará a atingir sua meta.

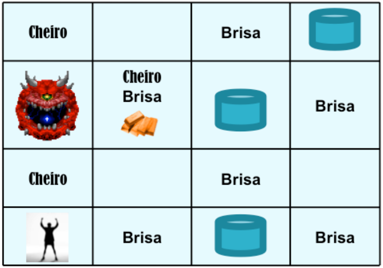
Abordagem declarativa

Essa análise independe de como o taxi funciona no nível de implementação – se seu conhecimento geográfico é implementado como grafos ou mapas de pixels, ou ainda se ele raciocina manipulando strings de símbolos ou propagando sinais em uma rede de neurônios.

Mundo do Wumpus

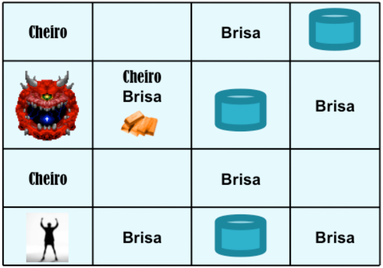
O Mundo do Wumpus

- é uma caverna que consiste em salas conectadas por passagens;
- à espreita em algum lugar está o Wumpus – um monstro que devora qualquer guerreiro que entrar em sua sala;
- o Wumpus pode ser atingido por uma flecha disparada por um agente, mas o agente só tem uma flecha;
- algumas salas contêm poços sem fundo nos quais cairá qualquer um que vagar por ela – com exceção do Wumpus que é muito grande para cair em um poço;
- nesse mundo é possível encontrar um monte de ouro.



Descrição do Mundo do Wumpus

- medida de desempenho – se há necessidade de otimizar as ações do agente "jogador" – então o mundo do Wumpus é visto como um jogo;
- ambiente e estado inicial – o agente inicia na posição [1,1], voltado para a direita, as posições dos demais objetos são escolhidas por acaso, sob uma probabilidade de ocorrência no caso dos poços, e não mudam durante a sessão de interação do agente com o ambiente;



Descrição do Mundo do Wumpus

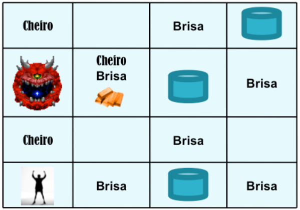
- atuadores
  - o agente pode se mover para frente, virar à esquerda ou à direita, morre se cair no poço ou se for pego pelo Wumpus;
  - mover-se para frente não terá efeito se houver uma parede;
  - a ação AGARRAR pode ser usada para levantar um objetivo que está no mesmo quadrado em que se encontra o agente;
  - a ação ATIRAR pode ser usada para disparar uma flecha em linha reta diante do agente. A flecha continua até achar o Wumpus ou uma parede. O agente só tem uma flecha.
- sensores
  - no quadrado contendo o Wumpus ou nos quadrados diretamente adjacentes (não na diagonal) o agente perceberá um cheiro ruim;
  - nos quadrados diretamente adjacentes a um poço, o agente perceberá uma brisa;
  - no quadrado onde está o ouro, o agente perceberá um resplendor;
  - quando caminhar para uma parede, o agente perceberá um impacto;
  - quando o Wumpus é morto, ele emite um grito triste que pode ser percebido em qualquer lugar na carverna

O agente explora o ambiente

Qual é a dificuldade????

A principal dificuldade para o agente é a sua ignorância inicial sobre a configuração do ambiente; superar essa ignorância parece exigir um raciocínio lógico.

Jogando sem usar o conhecimento prévio que temos do ambiente (estamos vendo o ambiente completo mas devemos desconsiderar isso e considerar apenas as regras do jogo)!

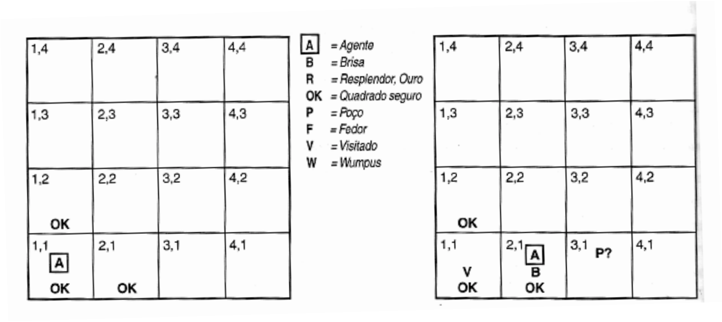


A base de conhecimento inicial do agente contém as regras do ambiente; em particular, ele sabe que está em [1,1] e que [1,1] é um quadrado seguro.

Evolução do conhecimento do agente

- a primeira percepção do agente é [nada, nada, nada, nada, nada], a partir da qual o agente conclui que os quadrados vizinhos ao que ele está são seguros;
- a partir do fato que não há nenhum cheiro ou brisa em [1,1], o agente pode deduzir que [1,2] e [2,1] estão livres dos perigos;
- suponha que o agente decida se mover para frente até [2,1];
- o agente detecta uma brisa em [2,1] e portanto deve haver um poço em um dos quadrados vizinhos. O poço não pode estar em [1,1], de acordo com as regras do jogo e, portanto, deve haver um poço em [2,2], [3,1] ou em ambos;
- neste momento, existe apenas um quadrado conhecido para onde o agente pode se mover com segurança e que ainda não foi visitado. Deste modo, o agente virará e retornará a [1,1] e depois prosseguirá para [1,2];
- o fedor em [1,2] significa que deve haver um Wumpus por perto. Pelas regras do jogo, o Wumpus não pode estar em [1,1]. Ele também não pode estar em [2,2], ou o agente teria detectado um cheiro quando estava em [2,1];
- o agente estão **deduz** que o Wumpus está em [1,3];
- além disso, a falta de uma brisa em [1,2] implica que não existe poço em [2,2]. Sabendo que não existe poço nem Wumpus em [2,2] o agente sabe que é OK se mover para lá.
- ...

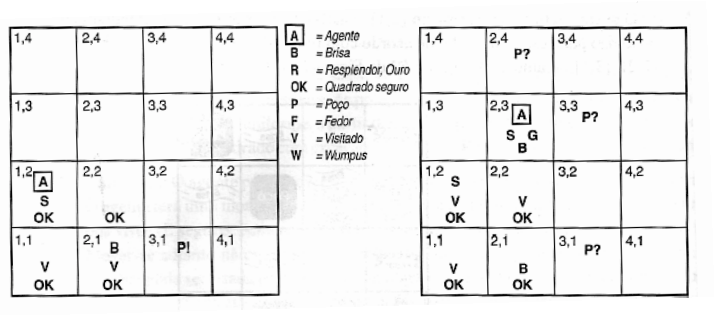
Representando graficamente



Percepções

- Na primeira figura a percepção é: [nada, nada, nada, nada, nada].
- Na segunda figura a percepção é: [nada, brisa, nada, nada, nada].

Representando graficamente



Percepções

- Na primeira figura a percepção é: [cheiro, nada, nada, nada, nada].
- Na segunda figura a percepção é: [cheiro, brisa, resplendor, nada, nada].

Raciocínio lógico

Propriedade Fundamental do Raciocínio Lógico

Em cada caso no qual o agente tira uma conclusão a partir das informações disponíveis, essa conclusão tem a garantia de ser correta se as informações disponíveis estiverem corretas. **Essa é uma propriedade fundamental do raciocínio lógico.**

Lógica

Sintaxe

Bases de conhecimento consistem em um conjunto de sentenças. Essas sentenças são expressas de acordo com a **sintaxe** da linguagem de representação escolhida. Tal sintaxe especifica todas as sentenças bem-formadas.

Exemplo na aritmética

- $x + y = 4$: sentença bem formada;
- $x4y+=$: sentença mal formada;

O raciocínio envolve a geração e a manipulação dessas configurações.

Lógica

Semântica

A **semântica** está relacionada com o significado das sentenças. Em lógica, a semântica da linguagem define a **verdade** de cada sentença com relação a cada **mundo possível** ou **modelo**.

Exemplo na aritmética

- $x + y = 4$ é verdadeira em um mundo no qual $x = 2$ e $y = 2$;
- mas ela é falsa em um mundo no qual $x = 1$ e $y = 1$.

Em lógicas-padrão ou lógicas clássicas, toda sentença deve ser **verdadeira** ou **falsa** em cada mundo possível. Não existe nenhuma posição "intermediária".

Lógica

Exemplos:

1. há muitos mundos possíveis, por exemplo, diferentes ambientes para o jogo do Wumpus definem diferentes condições do jogo e portanto, cada um deles é um *mundo possível*.
2. cada um dos mundos possíveis fixa a verdade ou falsidade de cada sentença definida naquele contexto.

1. se x e y representam o número de mulheres e homens que se sentam em torno de uma mesa para jogar *poker*;
2. a sentença $x + y = 4$ é verdadeira quando há quatro jogadores ao todo.

Os modelos possíveis são todas as atribuições possíveis de números às variáveis x e y. Cada atribuição fixa a verdade de qualquer sentença da aritmética cujas variáveis são x e y.

Lógica

Consequência Lógica

O raciocínio lógico envolve a consequência lógica entre sentenças, ou seja, a ideia de que uma sentença decorre logicamente de outra sentença:

$\alpha \models \beta$ se e somente se, em todo modelo (mundo / ambiente) no qual α é verdadeira, β também é verdadeira. Ou, pode-se dizer que, se α é verdadeira, então β também **deve** ser verdadeira.

Na aritmética

A sentença $x + y = 4$ tem como consequência lógica a sentença $4 = x + y$, ou $x = 4 - y$ etc. Então, em qualquer modelo em que a primeira sentença é verdadeira, as outras duas também serão.

Uma base de conhecimento pode ser considerada uma declaração, e com isso podemos dizer que uma base de conhecimento tem como consequência lógica uma sentença.

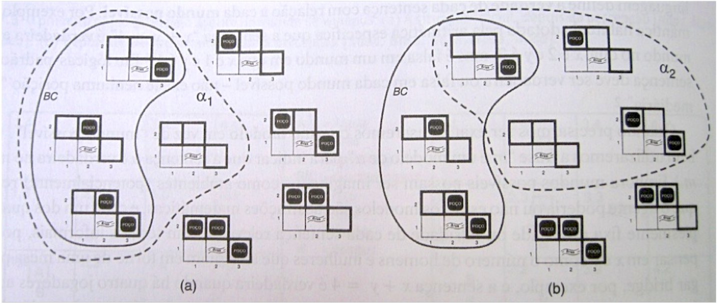
Lógica

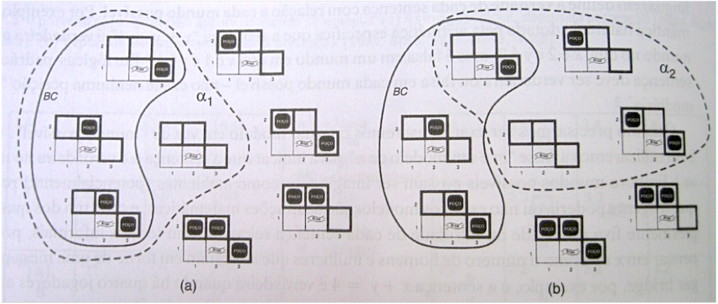
Exemplo

Considere a seguinte situação no mundo do Wumpus:

- o agente detectou **nada em [1,1]** e uma **brisa em [2,1]**; essas observações combinadas com o conhecimento que o agente já tem constituem a base de conhecimento (BC);
- o agente está interessado (entre outras coisas) em saber se os quadrados adjacentes [1,2], [2,2] e [3,1] contêm poços.

- cada um dos três quadrados ([1,2], [2,2], [3,1]) pode ou não conter um poço, e assim existem $2^3 = 8$ modelos possíveis;
- a BC é falsa em modelos que contradizem o que o agente sabe - por exemplo, a BC é falsa em qualquer modelo em que [1,2] contém um poço; ela também é falsa quando não há nenhum poço nas células adjacentes da célula [2,1];
- existem apenas três modelos no qual a BC é verdadeira, e
- considerando duas conclusões possíveis:
  - $\alpha_1 =$ "não existe nenhum poço em [1,2]";
  - $\alpha_2 =$ "não existe nenhum poço em [2,2]".





Por inspeção sobre os modelos, o raciocínio é:

- Em todo modelo no qual BC é verdadeira, α_1 também é verdadeira.
- Consequentemente, BC $\models \alpha_1$

e

- Em alguns modelos nos quais BC é verdadeira, α_2 é falsa.
- Consequentemente, BC $\models \alpha_2$ não vale, e o agente não pode concluir que não existe poço em [2,2].

Inferência Lógica

No exemplo anterior a definição de consequência lógica foi usada para derivar conclusões, ou seja, para conduzir a inferência lógica. O algoritmo usado é chamado **verificação de modelos**, porque enumera todos os modelos possíveis para verificar que α é verdadeira em todos os modelos em que BC é verdadeira.

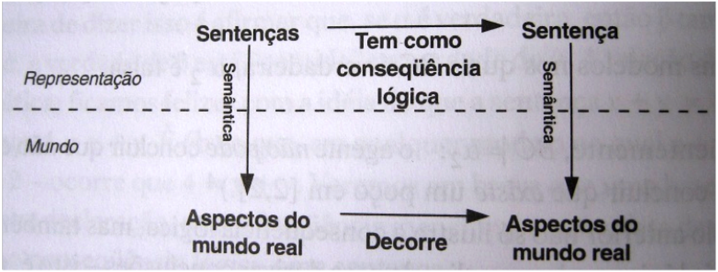
Algoritmo de inferência - propriedade de consistência

Um algoritmo de inferência que deriva apenas sentenças permitidas é dito consistente, ou seja, ele preserva a verdade. Um algoritmo de inferência não consistente inventa coisas à medida que prossegue. A verificação de modelos, quando aplicável, é um procedimento consistente.

Algoritmo de inferência - propriedade de completeza

Um algoritmo de inferência será completo se puder derivar qualquer sentença permitida. Para problemas finitos, um exame sistemático é completo. O problema está no espaço de busca infinito. Felizmente, existem procedimentos de inferência completos para lógicas que são suficientemente expressivas.

Inferência Lógica



Lógica Proposicional - **sintaxe**

A sintaxe define as sentenças permitidas.

- sentenças atômicas: elementos sintáticos indivisíveis – um único símbolo proposicional.
- sentenças complexas: construídas a partir de sentenças mais simples com a utilização de conectivos lógicos.

Sentenças atômicas

- Cada símbolo proposicional representa uma proposição que pode ser **verdadeira** ou **falsa**.
- Nomes em letras maiúsculas são usados para indicar os símbolos: P, Q, R ...
- Nomes são arbitrários e geralmente minemônicos: $\mathbf{W}_{1,3}$ é usado para representar a proposição de que o Wumpus está na posição [1,3].
- Existem dois símbolos proposicionais com significado fixo: VERDADEIRO é a proposição sempre verdadeira e FALSO é a proposição sempre falsa.

Lógica Proposicional - **sintaxe**

Existem cinco conectivos de uso comum

- $\neg$ Uma sentença como $\neg W_{1,3}$ é chamada de negação de $W_{1,3}$. Um literal é uma sentença atômica (um **literal positivo**) ou uma sentença atômica negada (**literal negativo**)
- $\wedge$ Um sentença cujo principal conectivo é $\wedge$, como $W_{1,3} \wedge P_{3,1}$, é chamada de **conjunção**; suas partes são os elementos da conjunção.
- $\vee$ Um sentença cujo principal conectivo é $\vee$, como $(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$, é chamada de **disjunção** dos disjuntos $W_{1,3} \wedge P_{3,1}$ e $W_{2,2}$
- $\Rightarrow$ Uma sentença como $(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$ é chamada **implicação** (ou condicional). Sua **premissa** ou **antecedente** é $(W_{1,3} \wedge P_{3,1})$, e sua **conclusão** ou **consequente** é $\neg W_{2,2}$. As implicações também são conhecidas como regras ou declações **se-então**.
- $\Leftrightarrow$ A sentença $W_{1,3} \Leftrightarrow \neg W_{2,2}$ é uma **bicondicional**.

Ordem de precedência e ambiguidade

A ordem de precedência é $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$. A precedência não resolve todas ambiguidades e parênteses precisam ser usados.

Lógica Proposicional - **semântica**

A semântica define as regras para determinar a verdade de uma sentença com respeito a um modelo específico. Em lógica proposicional, um modelo simplesmente fixa o valor verdade - *verdadeiro* ou *falso* - para todo símbolo proposicional.

Exemplo

Se as sentenças na base de conhecimento fizerem uso dos símbolos proposicionais $P_{1,2}$, $P_{2,2}$, $P_{3,1}$, então um modelo possível será: $m_1 = \{ P_{1,2} = \textit{falsa}, P_{2,2} = \text{falsa}, P_{3,1} = \textit{verdadeira} \}$

A semântica da lógica proposicional deve especificar como calcular o valor verdade de qualquer sentença, dado um modelo. Então, se sentenças são formadas a partir de sentenças atômicas combinadas com conectivos, é preciso especificar como calcular a verdade das sentenças atômicas e daquelas formadas com cada um dos conectivos.

Lógica Proposicional - **semântica**

Para sentenças atômicas

- VERDADEIRO é verdadeiro em todo o modelo e FALSO é falso em todo modelo.
- O valor-verdade de todos os outros símbolos proposicionais deve ser especificado no modelo. Por exemplo, no modelo m_1, $P_{1,2}$ é falsa.

Para sentenças complexas

- Para qualquer sentença *s* e qualquer modelo *m*, a sentença $\neg s$ é verdade em *m* se e somente se *s* é falso em *m*.
- ... **tabela verdade**.

Lógica Proposicional - **semântica**

Tabela verdade para os cinco conectivos lógicos.

P	Q	¬P	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
falso	falso	verdadeiro	falso	falso	verdadeiro	verdadeiro
falso	verdadeiro	verdadeiro	falso	verdadeiro	verdadeiro	falso
verdadeiro	falso	falso	falso	verdadeiro	falso	falso
verdadeiro	verdadeiro	falso	verdadeiro	verdadeiro	verdadeiro	verdadeiro

Considere: $m_1 = \{ P_{1,2} = \textit{falsa}, P_{2,2} = \text{falsa}, P_{3,1} = \textit{verdadeira} \}$

A sentença $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$ avaliada em m_1 resulta em: $\textit{verdadeiro} \wedge (\textit{falso} \vee \textit{verdadeiro}) = \textit{verdadeiro} \wedge \textit{verdadeiro} = \textit{verdadeiro}$.

Lógica Proposicional - **semântica**

Uma base de conhecimento consiste em um conjunto de sentenças. Uma base de conhecimento lógica é uma conjunção dessas sentenças representadas com lógica. Podemos representar a base de conhecimento do agente do mundo do Wumpos por meio de um conjunto de sentenças em lógica proposicional.

Analise as duas sentenças

- $\text{Brisa}_{1,1} \Leftrightarrow (\text{Poço}_{1,2} \vee \text{Poço}_{2,1})$
- $\text{Brisa}_{1,1} \Rightarrow (\text{Poço}_{1,2} \vee \text{Poço}_{2,1})$

A primeira sentença expressa melhor as condições do mundo do Wumpus. A segunda regra é verdadeira também, mas é incompleta. Ela não elimina modelos em que $\text{Brisa}_{1,1}$ é falsa e $\text{Poço}_{1,2}$ é verdadeira.

Pense! No mundo do Wumpus:

- se um quadrado tem uma brisa, então um ou mais quadrados vizinhos tem um poço; e
- se um quadrado tem um poço, então os quadrados vizinhos tem uma brisa.

Uma base de conhecimento simples

Considere o mundo do Wumpus parcial, ou seja, considere um mundo do Wumpus apenas com poços. Considere o seguinte vocabulário de símbolos proposicionais. Para cada i,j:

- Seja $P_{i,j}$ *verdadeira* se existe um poço em $[i,j]$.
- Seja $B_{i,j}$ *verdadeira* se existe um brisa em $[i,j]$.

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 : $B_{2,1}$.

obs.1: Os fatos referentes à célula 1,1 fazem parte da definição do estado inicial do problema.

obs.2: A regra sobre brisa em uma célula precisa ser declarada para todas as célula pois vale em todas as células do mundo do Wumpus.

obs.3: O fato referente à célula 2,1 vem da visita do agente à célula 2,1.

Um pouco sobre equivalência lógica

Duas sentenças α e β são logicamente equivalentes se são verdadeiras no mesmo conjunto de modelos (ou $\alpha \Leftrightarrow \beta$)

- comutatividade de $\wedge$: $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$
- comutatividade de $\vee$: $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$
- associatividade de $\wedge$: $(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$
- associatividade de $\vee$: $(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$
- eliminação de negação dupla: $\neg(\neg\alpha) \equiv \alpha$
- contraposição: $(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$
- eliminação de implicação: $(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$
- eliminação de bi-condicional: $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$
- de Morgan: $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$
- de Morgan: $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$
- distributividade de $\wedge$ sobre $\vee$: $(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$
- distribuitividade de $\vee$ sobre $\wedge$: $(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

Padrões de raciocínio para gerar cadeias de conclusões

Modus Ponens

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

Modus Tollens

$$\frac{\alpha \Rightarrow \beta, \neg\beta}{\neg\alpha}$$

Silogismo Hipotético

$$\frac{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}$$

Eliminação de e

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

Todas as equivalências lógicas do slide 27 podem ser usadas como regras de inferência.

Usando as regras de inferência no mundo do Wumpus

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 : $B_{2,1}$.

Inserindo novas sentenças à BC e provando $\neg P_{1,2}$

Aplicando a eliminação de bicondicional a R_2 obtém-se:

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

Usando as regras de inferência no mundo do Wumpus

Inserindo novas sentenças à BC e provando $\neg P_{1,2}$

Aplicando eliminação de E em R_6 obtém-se:

R_7 : $((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$.

Inserindo novas sentenças à BC e provando $\neg P_{1,2}$

Aplicando a contraposição em R_7, obtém-se:

R_8 : $(\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$.

Inserindo novas sentenças à BC e provando $\neg P_{1,2}$

Aplicando Modus Ponens com R_8 e R_4, obtém-se:

R_9 : $(\neg (P_{1,2} \vee P_{2,1})$.

Usando as regras de inferência no mundo do Wumpus

Inserindo novas sentenças à BC e provando $\neg P_{1,2}$

Aplicando de Morgan, gera-se a conclusão:

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

Ainda aplicaríamos a Eliminação de E para concluir (ou provar) que não existe poço em $P_{1,2}$ e não existe poço em $P_{2,1}$.

Resolução

As regras apresentadas até agora são consistentes mas para garantir completeza aos algortimos de inferência ainda é necessário o conceito de resolução.

Resolução no mundo do Wumpus

... o agente retorna de [2,2] para [1,1] e então vai para [1,2] onde percebe um fedor mas nenhuma brisa:

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

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

Resolução no mundo do Wumpus

... seguindo um processo de aplicação das regras de equivalência (**faça, como exercício**), obtém-se

R_{13} : $\neg P_{2,2}$.

R_{14} : $\neg P_{1,3}$.

R_{15} : $P_{1,1} \vee P_{2,2} \vee P_{1,3}$.

R_{16} : $P_{1,1} \vee P_{1,3}$.

Resolução

Em linguagem comum:

- se existe um poço em [1,1], [2,2] ou [3,1] e não existe poço em [2,2], então ele está em [1,1] ou [3,1].

Resolução no mundo do Wumpus

De modo semelhante, o literal $\neg P_{1,1}$ em R_1 se resolve com o literal $P_{1,1}$ em R_{16}, e gera

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

Inferência por prova de teorema

O mecanismo de inferência que acabamos de ver é conhecido como Prova de Teorema Proposicional. Ele pode ser implementado por um procedimento de busca. As definições gerais para modelar o procedimento como um problema de busca são:

- Estado inicial: uma base de conhecimento inicial
- Ações: o conjunto de ações consiste de todas as regras de inferência/equivalência/resolução aplicada a todas as sentenças da base
- Resultado: o resultado de uma ação é adicionar a setença na base de conhecimento
- Meta: o estado que contém a sentença que estamos tentando provar

Encadeamento para frente e para trás

Basedo em implicações cuja premissa é uma conjunção de literais positivos e cuja conclusão é um único literal positivo (derivados de cláusulas de HORN – disjunção de literais dos quais no máximo um é positivo).

Exemplo: $(\neg L_{1,1} \vee \neg brisa \vee B_{1,1})$

Cláusulas de Horn podem ser escritas da seguinte forma: $(L_{1,1} \wedge brisa) \Rightarrow B_{1,1}$

* $L_{1,1}$: significa que o agente está na célula [1,1]
* *brisa* : significa que o sensor de brisa captou uma brisa

Exercício: mostrar a equivalência afirmada neste slide

Encadeamento para frente e para trás

Encadeamento para frente

- Determina se um único símbolo proposicional - a consulta - é permitido por uma base de conhecimento de cláusulas de Horn.
- Ele começa de fatos conhecidos (literais positivos).
- Se todas as premissas de uma implicação foram conhecidas, sua conclusão será acrescentada ao conjunto de fatos conhecidos.
- Esse processo continua até a consulta ser acrescentada ou até não ser possível fazer inferências adicionais.

Encadeamento para frente e para trás

Encadeamento para trás

- Se a consulta (um símbolo proposicional) é reconhecida como verdadeira, não é necessário nenhum trabalho.
- Caso contrário, o algoritmo encontra as implicações na base de conhecimento que geram a conclusão (símbolo propocional da consulta).
- Se for possível demonstrar que todas as premissas de uma dessas implicações são verdadeiras (por encadeamento para trás), então a conclusão (consulta) é verdadeira.

Encadeamento para frente e para trás

Uma base de conhecimento:

- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Uma consulta: Q?

Inferência

O objetivo da inferência lógica, neste contexto, é decidir se BC $\models \alpha$. Por exemplo, $P_{2,2}$ é permitida?

Possibilidades

- Algoritmo de verificação de modelos: Enumerar todos os modelos e verificar se α é verdadeira em todo modelo no qual BC é verdadeira em todo modelo no qual BC é verdadeira.
- Prova de teorema: aplica-se as regras de inferência/equivalência/resolução para chegar à sentença desejada ou à sua negação.
- Encadeamento para frente e para trás: execução de etapas de inferência lógica óbvias e fáceis de acompanhar (pelos seres humanos).

Profa. Dra. Sarajane Marques Peres
Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Sala 320-A - Bloco I1
sarajane@usp.br `www.each.usp.br/sarajane`