

# Representação do Conhecimento e Raciocínio, Agentes baseados em LPO

# KR e Lógica de Primeira Ordem (LPO)

- É um formalismo de referência para representação de conhecimento
  - o mais estudado e o melhor formalizado
- Satisfaz em grande parte os seguintes critérios:
  - adequação representacional
    - permite representar o mundo (expressividade)
  - adequação inferencial
    - permite inferência
  - eficiência aquisicional
    - facilidade de adicionar conhecimento
  - modularidade

# Sistema Formal em LPO

## Cálculo

= Cálculo de Predicados

Linguagem  
= LPO

Regras de derivação  
= regras de inferência

sintaxe + semântica

## Teoria

### Axiomas

= fatos +  
regras gerais

### Teoremas

= fatos derivados  
(axiomas + regras  
de inferência)

**Base de Conhecimento** = fatos e regras gerais do domínio (só axiomas)

**Memória de Trabalho** = fatos particulares à instância do problema e fatos derivados (só fatos)

**Máquina de Inferência** = regras de inferência (de derivação)

# LPO: Sintaxe

Sentença  $\rightarrow$  SentençaAtômica | Sentença Conectivo Sentença  
| Quantificador Variável,... Sentença  
|  $\neg$  Sentença | (Sentença)

SentençaAtômica  $\rightarrow$  Predicado(Termo,...) | Termo = Termo | Verdade | Falso

Termo  $\rightarrow$  Função(Termo,...) | Constante | Variável

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

Quantificador  $\rightarrow$   $\forall$  |  $\exists$

Constante  $\rightarrow$  A | X | João | ...

Variável  $\rightarrow$  a | x | s | ...(letras minúsculas)

Predicado  $\rightarrow$  Vermelho | Tem-cor | IrmãoDe | ...

Função  $\rightarrow$  Mãe | MelhorAmigo | ...

# LPO: Semântica

- Constantes e Variáveis denotam Objetos:
  - ex. João, aluno, cadeira, estrela...
  - uma interpretação especifica a que objeto no mundo cada constante e cada variável livre se refere.
  - o mesmo objeto pode ser referenciado por mais de um símbolo:
    - ex. “Vênus” e “A estrela Dalva” referem-se ao mesmo objeto (planeta) no universo.

# LPO: Semântica

- **Predicados** denotam **Propriedades e Relações**:
  - uma **interpretação** especifica a que propriedade (predicado unário) ou relação no mundo o predicado se refere:
    - ex., **irmão** = predicado binário que se refere à relação de irmandade: **irmão**(Caetano, Betânia )
    - ex. **Feio**(Maguila) - faz referência à propriedade ser feio

# LPO: Semântica

- Funções denotam Relações Funcionais:
  - ligam um objeto (ou mais) a um único objeto no mundo
- Uma interpretação especifica
  - que relação funcional no mundo é referida pelo símbolo da função, e
  - que objetos são referidos pelos termos que são seus argumentos
    - Termos denotam Objetos:
      - são constantes, variáveis ou funções.

# LPO: Semântica

- Funções denotam Relações Funcionais:
  - Exemplos:
    - 1) Mãe(Roberto Carlos) -> LadyLaura
      - função que devolve o nome da mãe do seu argumento
    - 2) Nota(Zezinho) = 9
      - devolve a nota do argumento Zezinho



# LPO: Semântica

- **Sentenças Atômicas:** predicados + termos
  - o **valor-verdade** de uma sentença depende da **interpretação** e do **mundo**.
    - ♦ Irmão(Caetano, Betânia)
      - termos simples
    - ♦ Casados(PaiDe(Caetano),MãeDe(Betânia))
      - termos complexos

# LPO: Semântica

- **Sentenças Complexas:**
  - usam conectivos e quantificadores
  - a semântica dessas sentenças é atribuída da mesma maneira que na lógica proposicional:
    - ◆ semântica dos conectivos + valor-verdade das sentenças individuais.
    - ◆  $\text{Irmão}(\text{Caetano}, \text{Betânia}) \wedge \text{Filho}(\text{Zeca}, \text{Caetano}) \Rightarrow \text{Tia}(\text{Betânia}, \text{Zeca})$

# LPO: Semântica

- Igualdade:
  - declara que dois termos se referem ao mesmo objeto
    - Mãe(Roberto Carlos) = Lady Laura
  - pode ser visto como a relação identidade...
- Exemplo
  - Para dizer que Huguinho tem pelo menos dois irmãos (Luizinho e Zezinho), escreve-se:
  - $\exists x, y \text{ irmão}(\text{Huguinho}, x) \wedge \text{irmão}(\text{Huguinho}, y) \wedge \neg(x = y)$

# LPO: semântica dos quantificadores

- **Universal** ( $\forall$ )
  - **conjunção** sobre o universo de objetos
    - $\forall x P(x)$  é **verdade** sse  $P$  é verdade para **todos** os objetos no mundo
  - Ex.  $\forall x \text{ Gato}(x) \Rightarrow \text{Mamífero}(x)$ 
    - ♦ o valor-verdade é dado pela semântica do quantificador universal e do conectivo  $\Rightarrow$

# LPO: semântica dos quantificadores

- Existencial ( $\exists$ )
  - **disjunção** sobre o universo de objetos  
 $\exists x P(x)$  é **verdade** sse  $P$  é verdade para **algum** objeto no mundo.
  - Ex.  $\forall x \exists y \text{ pessoa}(x) \wedge \text{mãe}(y,x)$
- Correspondência entre os dois quantificadores
  - $\forall x \text{ Gosta}(x, \text{Banana}) \Leftrightarrow \neg \exists x \neg \text{Gosta}(x, \text{Banana})$

# LPO: Leis de De Morgan

- Leis de De Morgan: Equivalência entre sentenças quantificadas e entre sentenças não quantificadas

$$\models \forall(x) \neg P \equiv \neg \exists(x) P$$

$$\neg P \wedge \neg Q \equiv \neg (P \vee Q)$$

$$\models \neg \forall(x) P \equiv \exists(x) \neg P$$

$$\neg (P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\models \forall(x) P \equiv \neg \exists(x) \neg P$$

$$P \wedge Q \equiv \neg (\neg P \vee \neg Q)$$

$$\models \exists(x) P \equiv \neg \forall(x) \neg P$$

$$P \vee Q \equiv \neg (\neg P \wedge \neg Q)$$

- Moral da história:
  - Não precisamos de ambos os quantificadores  $\forall$  e  $\exists$  nem de ambos os conectivos  $\wedge$  e  $\vee$  ao mesmo tempo!
  - Útil para melhorar a eficiência da inferência
    - ex. cláusulas de Horn (PROLOG)

# Propriedades da Inferência Lógica

- Corretude
  - gera apenas sentenças válidas
- Composicionalidade
  - o significado de uma sentença é função do de suas partes
- Monotonicidade
  - Tudo que era verdade continua sendo depois de uma inferência
- Localidade
  - inferência apenas com comparações locais (porção da BC).
- Localidade e composicionalidade ---> modularidade ---> reusabilidade e extensibilidade

# Sistemas baseados em LPO



# Sistemas baseados em LPO

Representando sentenças no mundo:

Pedro possui um cachorro.

Todo dono de cachorro é um protetor dos animais.

Nenhum protetor dos animais mata um animal.

Representando sentenças na Lógica:

$\exists x \text{ cachorro}(x) \wedge \text{possui}(\text{Pedro}, x)$

$\forall x \exists y (\text{cachorro}(y) \wedge \text{possui}(x, y)) \Rightarrow \text{protetorAnimais}(x)$

$\forall x \text{ protetorAnimais}(x) \wedge \forall y \text{ animal}(y) \Rightarrow \neg \text{mata}(x, y)$

# Sistemas baseados em LPO

- Base de Conhecimento
  - fatos e regras básicos, gerais, permanentes
    - $\forall(x,z) \text{ Avó}(x,z) \Leftrightarrow \exists(y) \text{ Mãe}(x,y) \wedge (\text{Mãe}(y,z) \vee \text{ Pai}(y,z))$
- Memória de Trabalho
  - fatos particulares à instância do problema
    - $\text{Pai}(\text{Caetano}, \text{Zeca}), \text{Mãe}(\text{Canô}, \text{Caetano})$
  - e fatos derivados
    - $\text{Avó}(\text{Canô}, \text{Zeca})$
- Máquina de Inferência
  - regras de inferência

# Sistemas baseados em LPO

- Primeiros passos
  1. Armazenar as regras da BC na máquina de inferência (MI) e os fatos na memória de trabalho (MT)
  2. Adicionar os dados iniciais à memória de trabalho

# Agentes baseados em LPO

- função  $\text{Agente-BC}(\text{percepção})$  retorna uma *ação*  
     $\text{Tell}(\text{MT}, \text{Percepções-Sentença}(\text{percepção}, t))$   
     $\text{ação} \leftarrow \text{Ask}(\text{MT}, \text{Pergunta-Ação}(t))$   
     $\text{Tell}(\text{MT}, \text{Ação-Sentença}(\text{ação}, t))$   
     $t \leftarrow t + 1$   
    retorna *ação*
- **Onde...**
  - MT – memória de trabalho
  - função Pergunta-Ação cria uma *query* como:  $\exists a \text{ Ação}(a, 6)$
  - função ASK devolve uma lista de instanciações:  
     $\{a / \text{Pegar}\}$  - Pegar é atribuída à variável ação.
  - função TELL grava a ação escolhida na memória de trabalho.

# Agentes baseados em LPO

- Como a função *ASK* responde as *queries*
  - Quantificador  $\forall$ : a resposta é booleana
    - $\text{ASK}(\text{BC}, \text{Irmã}(\text{Betânia}, \text{Caetano})) \rightarrow \text{true}$
    - $\text{ASK}(\text{BC}, \forall x (\text{Irmã}(x, \text{Caetano}) \wedge \text{Cantora}(x))) \rightarrow \text{false}$
  - Quantificador  $\exists$ : a resposta é uma lista de instâncias/substituições de variáveis - *binding*
    - $\text{ASK}(\text{BC}, \exists x \text{Irmã}(x, \text{Caetano})) \rightarrow \{\mathbf{x/Betânia}, \mathbf{x/Irene}\}$
    - $\text{ASK}(\text{BC}, \exists x (\text{Irmã}(x, \text{Caetano}) \wedge \text{Cantora}(x))) \rightarrow \{\mathbf{x/Betânia}\}$

# Hipótese do Mundo Fechado

- Tudo que não estiver presente na base é considerado falso
- Isto simplifica (reduz) a BC
  - Ex. Para dizer que a população dos países Nova Zelândia, África do Sul, Irlanda e França gostam do jogo Rugby, não precisa explicitamente dizer que os outros não gostam...

# Um Exemplo de Construção de BC

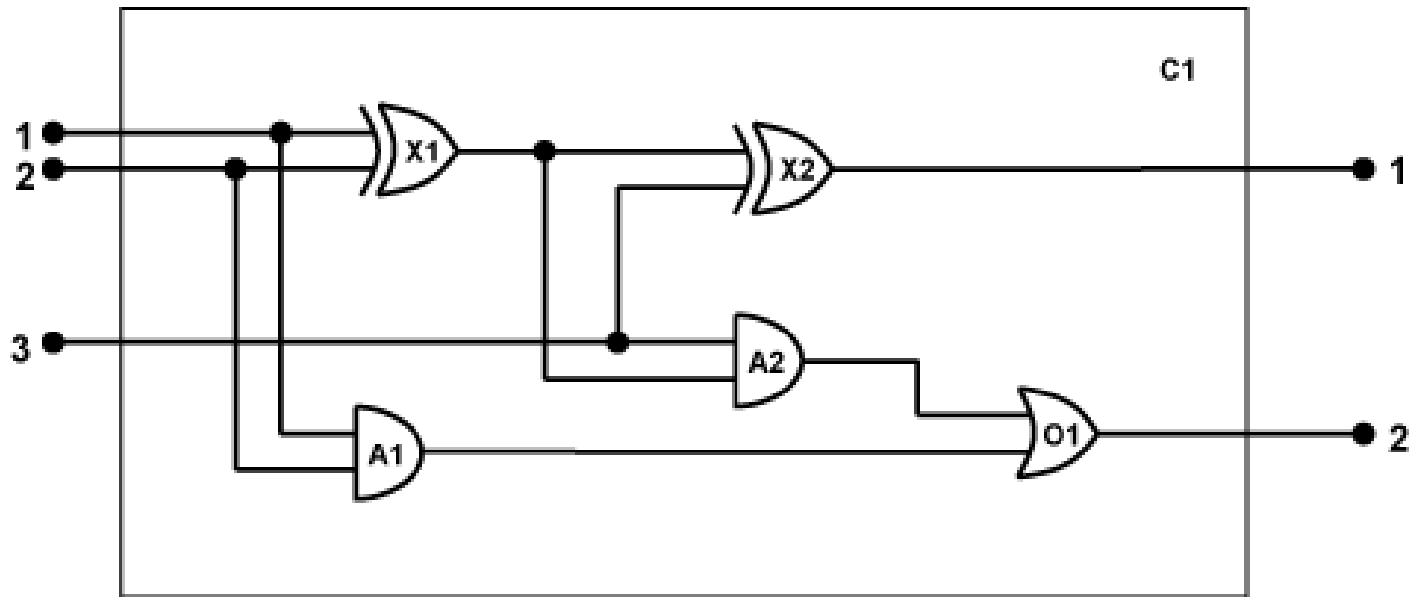
Do livro AIMA

# Construindo uma BC

- 1) Decida sobre o *que* falar
- 2) Escolha o *vocabulário* de predicados, funções e constantes  
**(*Ontologia do Domínio*)**
- 3) Codifique o *conhecimento genérico* sobre o domínio  
**(*axiomas*)**  
$$\forall x,y,z \text{ Americano}(x) \wedge \text{Arma}(y) \wedge \text{Nação}(z) \wedge \text{Hostil}(z) \wedge \text{Vende}(x,z,y) \Rightarrow \text{Criminoso}(x)$$
- 4) Codifique uma *descrição* de uma *instância* específica do problema: *Nação(Cuba), Nação(USA)*
- 5) Proponha *questões* para o procedimento de inferência e obtenha *respostas*: *West é criminoso?*



# Um Exemplo: Circuitos Digitais



- Objetivo:
  - determinar se o circuito está de acordo com sua especificação (o circuito acima é um *somador*)
  - responder a perguntas sobre o valor da corrente em qualquer ponto do circuito

# Decida sobre o que falar

- Para alcançar o objetivo, é relevante falar sobre
  - circuitos, terminais, sinais nos terminais, conexões entre terminais
- Para determinar quais serão esses sinais, precisamos saber sobre:
  - portas e tipos de portas: AND, OR, XOR e NOT
- Não é relevante falar sobre:
  - fios, caminhos dos fios, cor e tamanho dos fios, etc.

# Decida qual vocabulário usar

- Usado para nomear objetos e relações do domínio com funções, predicados e constantes
  - constantes
    - distinguir as portas:  $X1, X2...$
    - distinguir os tipos de porta:  $AND, OR, XOR...$
  - funções e predicados
    - tipo de uma porta:  
 $Tipo(X1) = XOR, Tipo(X1, XOR), XOR(X1)$
    - indicar entradas e saídas:  
 $Out(1, X1), In(1, X2)$
    - indicar conectividade entre portas:  
 $Conectado(Out(1, X1), In(1, X2))$

# Codifique regras genéricas

(1) Dois terminais conectados têm o mesmo sinal:

$$\forall t1, t2 \text{ Conectado}(t1, t2) \Rightarrow \text{Sinal}(t1) = \text{Sinal}(t2)$$

(2) O sinal de um terminal é *On* ou *Off* (nunca ambos)

$$\forall t \text{ Sinal}(t) = \text{On} \vee \text{Sinal}(t) = \text{Off}, \text{ On} \neq \text{Off}$$

(3) *Conectado* é um predicado comutativo

$$\forall t1, t2 \text{ Conectado}(t1, t2) \Leftrightarrow \text{Conectado}(t2, t1)$$

(4) Uma porta *OR* está *On* sse qualquer das suas entradas está *On*:

$$\begin{aligned} \forall g \text{ Tipo}(g) = \text{OR} &\Rightarrow \text{Sinal}(\text{Out}(1,g)) = \text{On} \\ \Leftrightarrow &\quad \exists n \text{ Sinal}(\text{In}(n,g)) = \text{On} \end{aligned}$$

# Codifique a instância específica

- Portas:

*Tipo(X1) = XOR   Tipo(X2) = XOR*

*Tipo(A1) = AND   Tipo(A2) = AND*

*Tipo(O1) = OR*

- Conexões:

*Conectado(Out(1,X1),In(1,X2))*

*Conectado(Out(1,X1),In(2,A2))*

*Conectado(Out(1,A2),In(1,O1)) . . .*

# Proponha questões ao mecanismo de Inferência

- Que entradas causam  $Out(1, C1) = Off$  e  $Out(2, C1) = On$ ?

$$\exists i1, i2, i3 \quad Sinal(In(1, C1)) = i1 \wedge$$

$$Sinal(In(2, C1)) = i2 \wedge Sinal(In(3, C1)) = i3 \wedge$$

$$Sinal(Out(1, C1)) = Off \wedge Sinal(Out(2, C1)) = On$$

- Resposta:

$$(i1 = On \wedge i2 = On \wedge i3 = Off) \vee$$

$$(i1 = On \wedge i2 = Off \wedge i3 = On) \vee$$

$$(i1 = Off \wedge i2 = On \wedge i3 = On)$$

# Agentes LPO para o Mundo do Wumpus

# Um Agente LPO para o Mundo do Wumpus

- ◆ Interface entre o agente e o ambiente:
  - sentença de percepções, que inclui as percepções e o tempo (passo) em que elas ocorreram
    - ◆ e.g.:
    - ◆ Percepção ([Fedor, Vento, Brilho, nada(~choque), nada(~grito)], 6)
- ◆ Ações do agente:
  - Girar(Direita), Girar(Esquerda), Avançar, Atirar, Pegar, Soltar e Sair das cavernas



# Um Agente LPO para o Mundo do Wumpus

- ◆ Três arquiteturas de Agentes baseados em LPO:
  - Agente reativo
  - Agente com Modelo do Mundo
  - Agente baseado em Objetivo

# Agente reativo baseado em LPO

- ◆ Possui regras ligando as seqüências de percepções a ações
  - Essas regras assemelham-se a reações
    - $\forall f,v,c,g,t \text{ Percepção}([f,v, \text{Brilho},c,g], t) \Rightarrow \text{Ação}(\text{Pegar}, t)$
- ◆ Essas regras dividem-se entre
  - Regras de (interpretação) da percepção
    - $\forall v,b,c,g,t \text{ Percepção}([\text{Fedor},v,b,c,g], t) \Rightarrow \text{Fedor}(t)$
    - $\forall f,b,c,g,t \text{ Percepção}([f,\text{Vento},b,c,g], t) \Rightarrow \text{Vento}(t)$
    - $\forall f,v,c,g,t \text{ Percepção}([f,v,\text{Brilho},c,g], t) \Rightarrow \text{Junto-do-Ouro}(t)$
    - . . .
  - Regras de ação
    - $\forall t \text{ Junto-do-Ouro}(t) \Rightarrow \text{Ação}(\text{Pegar}, t)$

# Limitações do agente reativo puro

- ◆ Como já vimos, um agente reativo puro nunca sabe quando parar
  - estar com o ouro e estar na caverna (1,1) não fazem parte da sua percepção
    - ◆ se pegou, esqueceu
  - esses agentes podem entrar em laços infinitos.
- ◆ Para ter essas informações, o agente precisa guardar uma representação do mundo.

# Agentes LPO com Estado Interno

## ◆ Guardando modelo interno do mundo (MT)

- sentenças sobre o estado atual do mundo
  - ◆ “agente está com o ouro”
- O modelo será atualizado quando
  - ◆ O agente receber novas percepções e realizar ações
  - ◆ ex. o agente pegou o ouro,...

## ◆ Questão

- Como manter, com simplicidade, o modelo do mundo corretamente atualizado?

# Representando Mudanças no Mundo

- ◆ Como representar as mudanças?
  - Ex., “O agente foi de [1,1] para [1,2]”
    1. Apagar da MT sentenças que já não são verdade
      - ◆ ruim: perdemos o conhecimento sobre o passado, o que impossibilita previsões de diferentes futuros.
    2. Cada estado é representado por uma BC/MT diferente:
      - ◆ ruim: pode explorar situações hipotéticas, porém não pode raciocinar sobre mais de uma situação ao mesmo tempo.
        - ex. “existiam buracos em (1,2) e (3,2)?”

# Cálculo Situacional

- ◆ Solução: Cálculo situacional !
  - uma maneira de escrever mudanças no tempo em LPO
  - Permite a representação de diferentes situações na mesma BC/MT
  
- ◆ Cap 10, pag. 329 do livro novo

# Cálculo Situacional

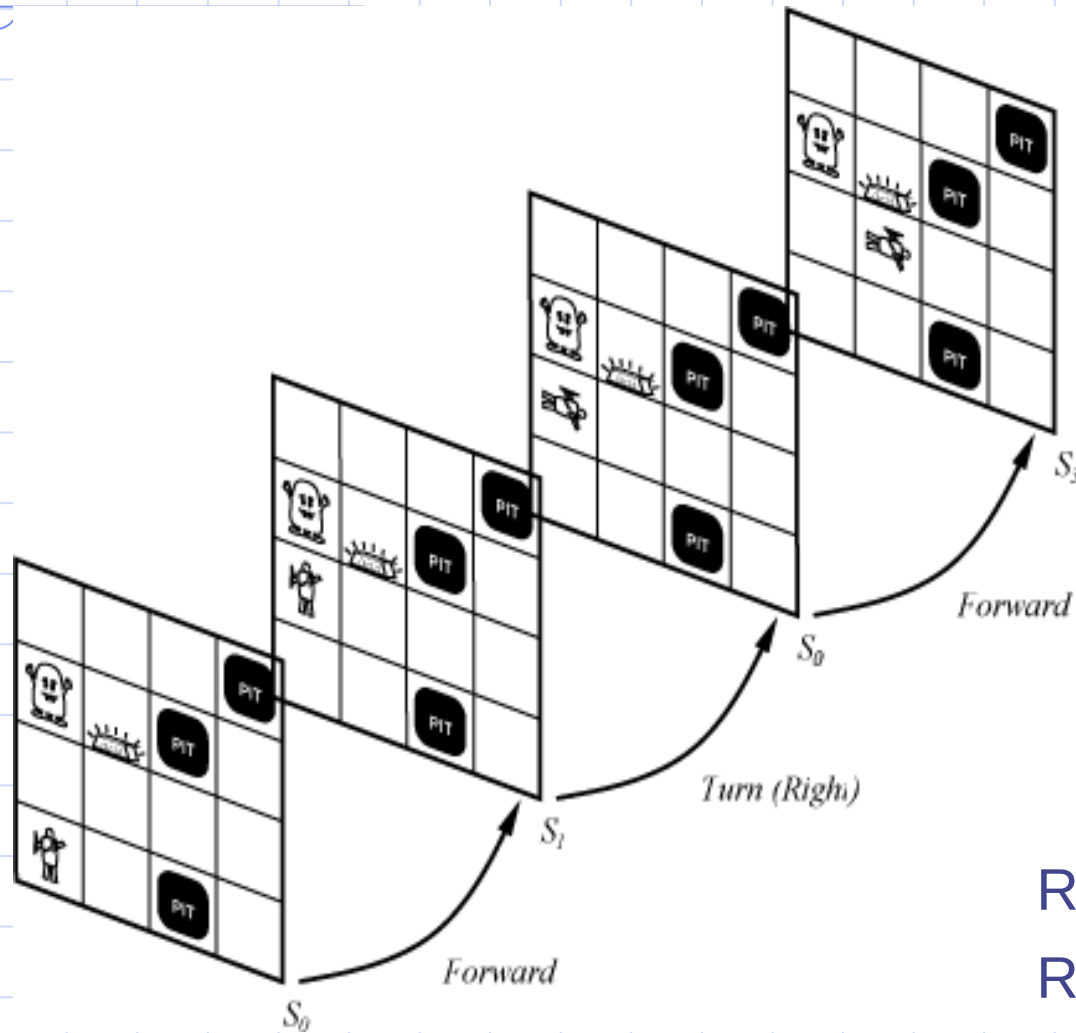
- ◆ Predicados que **mudam** com o tempo têm um argumento adicional de **situação** (tempo, turno)
  - Ao invés de  $Em(Agente, local)$
  - teremos  $Em(Agente, [1, 1], S0) \wedge Em(Agente, [1, 2], S1)$
- ◆ Predicados que denotam propriedades que não mudam com o tempo não necessitam de argumentos de situação
  - $Parede(0, 1)$  e  $Parede(1, 0)$

# Cálculo Situacional

- ◆ O mundo consiste em uma seqüência de situações
  - situação N  $\Rightarrow$  ação  $\Rightarrow$  situação N+1
- ◆ Utiliza uma função *Resultado* para representar as mudanças no mundo:
  - Resultado (ação, situação N) = situação N+1



# Exemplo de cálculo situacional



Resultado(Forward,  $S_0$ ) =  $S_1$

Resultado(Turn(Right),  $S_1$ ) =

$S_2$

Resultado(Forward,  $S_2$ ) =  $S_{341}$

# Representando Mudanças no Mundo

## Axiomas estado-sucessor

- ◆ Descrição completa de como o mundo evolui
  - uma coisa é verdade depois  $\Leftrightarrow$ 
    - [uma ação acabou de torná-la verdade
      - ∨ ela já era verdade e nenhuma ação a tornou falsa ]
  - Ex.  $\forall a - \text{ação}, o - \text{ouro}, sit \text{ Segurando}(o, \text{Resultado}(a,s))$ 
    - $\Leftrightarrow$
    - □□[(a = Pegar  $\wedge$  (Junto-do-ouro(s))
    - ∨ (Segurando (o,s)  $\wedge$  (a  $\neq$  Soltar)]
    - Obs.: Resultado(a,s) = s+1
- ◆ É necessário escrever uma axioma estado-sucessor para cada predicado que pode mudar seu valor no tempo.

# Representando Mudanças no Mundo do Wumpus

## Axiomas estado-sucessor

- ◆ O que muda com o tempo no mundo do Wumpus?
  - Pegar ouro, localização e orientação do agente
- ◆ Guardando localizações
  - O agente precisa lembrar por onde andou e o que viu
  - para poder deduzir onde estão os buracos e o Wumpus, e
  - para garantir uma exploração completa das cavernas

# Representando Mudanças no Mundo do Wumpus

## ◆ O agente precisa saber:

- **localização inicial** = onde o agente está

Em (Agente,[1,1],S0 )

- **orientação**: a direção do agente (em graus)

Orientação (Agente,S0 ) = 0

- **localização um passo à frente**: função de locais e orientações

$\forall x,y \text{ PróximaLocalização } ([x,y ],0) = [x+1,y]$

$\forall x,y \text{ PróximaLocalização } ([x,y ],90) = [x,y+1]$

$\forall x,y \text{ PróximaLocalização } ([x,y ],180) = [x-1,y]$

$\forall x,y \text{ PróximaLocalização } ([x,y ],270) = [x,y-1]$

# Representando Mudanças no Mundo do Wumpus

- ◆ A partir desses axiomas, pode-se deduzir que caverna está em frente ao agente “ag” que está na localização “loc”:

$$\forall \text{ ag, loc, s } \text{Em}(\text{ag, loc, s}) \Rightarrow \\ \text{localizaçãoEmFrente}(\text{ag, s}) = \\ \text{PróximaLocalização}(\text{loc, Orientação}(\text{ag, s}))$$

- ◆ Assim, a informação sobre a localização em frente ao agente em cada situação fica diretamente disponível na MT

# Representando Mudanças no Mundo do Wumpus

- ◆ Podemos também definir **adjacência**:

$\text{loc1, loc2 Adjacente (loc1, loc2)} \Leftrightarrow$

$\exists d \text{ loc1} = \text{PróximaLocalização (loc2, d)}$

- ◆ E detalhes geográficos do mapa:

$x, y \text{ Parede}([x, y]) \Leftrightarrow (x = 0 \vee x = 5 \vee y = 0 \vee y = 5)$

- ◆ Assim, informações sobre o mapa do ambiente ficam disponível na BC/MT

# Resultado das ações do agente sobre sua localização

## ◆ Axioma Estado-Sucessor

- avançar é a única ação que muda a localização do agente (a menos que haja uma parede)

$\Box \forall a, loc, ag, s \text{ Em}(ag, loc, \text{Resultado}(a, s)) \Leftrightarrow$

$\Box [(a = \text{Avançar} \wedge loc = \text{localizaçãoEmFrente}(ag, s) \wedge \neg \text{Parede}(loc))$   
 $\vee (\text{Em}(ag, loc, s) \wedge a \neq \text{Avançar})]$

# Resultado das ações do agente sobre sua orientação

## ◆ Axioma Estado-Sucessor

- girar é a única ação que muda a direção do agente

□  $\forall a, d, ag, s \text{ Orientação}(ag, \text{Resultado}(a, s)) = d \Leftrightarrow$

$[(a = \text{Girar(Direita)} \wedge d = \text{Mod}(\text{Orientação}(ag, s) - 90, 360))$

$\vee (a = \text{Girar(Esquerda)} \wedge d = \text{Mod}(\text{Orientação}(ag, s) + 90, 360))$

$\vee (\text{Orientação}(ag, s) = d \wedge \neg (a = \text{Girar(Direita)} \wedge a = \text{Girar(Esquerda)}))]$



# Deduzindo Propriedades do Mundo

- ◆ Agora que o agente sabe onde está em cada situação, ele pode associar **propriedades** aos **locais**:
  - $\forall ag, loc, s \text{ Em}(ag, loc, s) \wedge \text{Vento}(s) \Rightarrow \text{Ventilado}(loc)$
  - $\forall ag, loc, s \text{ Em}(ag, loc, s) \wedge \text{Fedor}(s) \Rightarrow \text{Fedorento}(loc)$
  - Observem que os predicados **Ventilado** e **Fedorento** não necessitam do argumento de situação
- ◆ Sabendo isto, o agente pode **deduzir**:
  - onde estão os buracos e o Wumpus, e
  - quais são as cavernas seguras (predicado *OK*).

# Tipos de regras

Que definem o tipo de sistema  
construído...

# Tipos de regras

## ◆ Regras Diacrônicas (do grego “através do tempo”)

- descrevem como o mundo evolui (muda ou não) com o tempo

$$\begin{aligned} &\forall x,s \text{ Presente}(x,s) \wedge \text{Portável}(x) \\ \Rightarrow &\text{Segurando}(x, \text{Resultado}(\text{Pegar}, s)) \end{aligned}$$

## ◆ Regras Síncronas

- relacionam propriedades na mesma situação (tempo).
- □  $\forall \text{loc}, s \text{ Em}(\text{Agente}, \text{loc}, s) \wedge \text{Vento}(s) \Rightarrow \text{Ventilado}(\text{loc})$
- possibilitam deduzir propriedades escondidas no mundo
- Existem dois tipos principais de regras síncronas:
  - ◆ Regras Causais e Regras de Diagnóstico.

# Regras síncronas causais

- ◆ Regras Causais assumem **causalidade**
  - algumas propriedades no mundo causam certas percepções.
  - Exemplos
    - ◆ as cavernas adjacentes ao Wumpus são **fedorentas**  
:
      - $\forall \text{ loc1, loc2, s } \text{Em}(\text{Wumpus}, \text{loc1}, \text{s}) \wedge \text{Adjacente}(\text{loc1}, \text{loc2}) \Rightarrow \text{Fedorento}(\text{loc2})$
    - ◆ **Se choveu, a grama está molhada**
  - Sistemas que raciocinam com regras causais são conhecidos como **Sistemas Baseados em Modelos**.

# Regras síncronas de diagnóstico

## ◆ Regras de Diagnóstico:

- Raciocínio abdutivo: supõe a presença de propriedades escondidas a partir das percepções do agente
- Ex., a ausência de fedor ou Vento implica que esse local e os adjacentes estão OK
  - $\forall \text{loc1,loc2,b,g,c,s} \text{ Percepção } ([\text{nada}, \text{nada}, \text{b,g,c}], \text{s}) \wedge$   
 $\text{Em } (\text{Agente}, \text{loc1}, \text{s}) \wedge \text{Adjacente}(\text{loc1}, \text{loc2}) \Rightarrow$   
 $\text{OK}(\text{loc2})$ 
    - ◆ se a grama está molhada, então é porque o aguador ficou ligado
- Sistemas que raciocinam com regras de diagnóstico são conhecidos como Sistemas de Diagnóstico

# Tipos de regras

## ◆ Atenção:

- Não se deve misturar numa mesma BC regras causais e de diagnóstico!!!
- se choveu é porque o aguador estava ligado...

# Sistema de Ação-Valor

Modularidade das Regras  
Adequação das regras

# Modularidade das Regras

- ◆ As regras que definimos até agora não são totalmente **modulares**
  - mudanças nas crenças do agente sobre algum aspecto do mundo requerem mudanças nas regras que lidam com outros aspectos que não mudaram
- ◆ Para tornar essas regras mais modulares, separamos fatos e regras sobre **ações** de fatos e regras sobre **objetivos**
  - assim, o agente pode ser “reprogramado” mudando-se o seu **objetivo** quando necessário



# Modularidade das Regras

- ◆ Ações descrevem como alcançar resultados.
- ◆ Objetivos descrevem a adequação (*desirability*) de estados resultado
  - não importando como foram alcançados.
- ◆ Assim, descrevemos a adequação das regras e deixamos que a máquina de inferência escolha a ação mais adequada

# Adequação das Regras

## ◆ Ações podem ser

- ótimas, boas, médias, arriscadas ou mortais.
- Escala, em ordem decrescente de adequação

## ◆ Assim, pode-se escolher a ação mais adequada para a situação atual

- meta regras que determinam a prioridade de execução das regras – desempate
- $\square \forall a,s \text{ Ótima}(a,s) \Rightarrow \text{Ação}(a,s)$
- $\square \forall a,s \text{ Boa}(a,s) \wedge (\neg \exists b \text{ Ótima}(b,s)) \Rightarrow \text{Ação}(a,s)$
- $\square \forall a,s \text{ Média}(a,s) \wedge (\neg \exists b (\text{Ótima}(b,s) \vee \text{Boa}(b,s))) \Rightarrow \text{Ação}(a,s)$
- $\square \forall a,s \text{ Arriscada}(a,s) \wedge (\neg \exists b (\text{Ótima}(b,s) \vee \text{Boa}(b,s) \vee \text{Média}(a,s))) \Rightarrow \text{Ação}(a,s)$

# Adequação das Regras

- ◆ Essas regras são gerais, e podem ser usadas em situações diferentes:
  - uma ação arriscada na situação S0
    - ◆ onde o Wumpus está vivo
  - pode ser ótima na situação S2
    - ◆ quando o Wumpus já está morto
- ◆ Sistema de Ação-Valor
  - Sistema baseado em regras de adequação
  - Não se refere ao que a ação faz, mas a quão desejável ela é.

# Sistema de Ação-Valor

- ◆ Prioridades do agente até encontrar o ouro:
  - ações ótimas: pegar o ouro quando ele é encontrado, e sair das cavernas.
  - ações boas: mover-se para uma caverna que está OK e ainda não foi visitada.
  - ações médias: mover-se para uma caverna que está OK e já foi visitada.
  - ações arriscadas: mover-se para uma caverna que não se sabe com certeza que não é mortal, mas também não é OK
  - ações mortais: mover-se para cavernas que sabidamente contêm buracos ou o Wumpus vivo.

# Agentes Baseados em Objetivos

- ◆ O conjunto de regras de adequação (ações-valores) é suficiente para prescrever uma boa estratégia de exploração inteligente das cavernas
  - quando houver uma seqüência segura de ações , ele acha o ouro
- ◆ Depois de encontrar o ouro, a estratégia deve mudar...
  - **novo objetivo**: estar na caverna (1,1) e sair.
    - $\forall s \text{ Segurando(ouro},s) \Rightarrow \text{LocalObjetivo}([1,1],s)$
- ◆ A presença de um **objetivo explícito** permite que o agente encontre uma seqüência de ações que alcançam esse objetivo

# Como encontrar seqüências de ações

## (1) Inferência:

- Idéia: escrever axiomas que perguntam à BC/MT uma seqüência de ações que com certeza alcança o objetivo.
- Porém, para um mundo mais complexo, isto se torna muito caro
  - ◆ como distinguir entre boas soluções e soluções mais dispendiosas (onde o agente anda “à toa” pelas cavernas)?

# Como encontrar seqüências de ações

## (2) Planejamento

- utiliza um sistema de raciocínio dedicado, projetado para raciocinar sobre ações e conseqüências para **objetivos diferentes**.

