

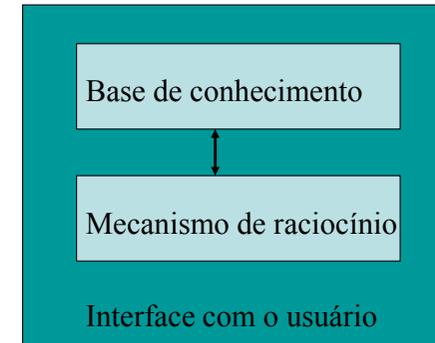
Motor de Inferência para Sistemas Especialistas baseados em Regras de Produção

prof. Luis Otavio Alvares

prof. Luis Otavio Alvares

Sistemas especialistas

Arquitetura geral:



prof. Luis Otavio Alvares

Motor de inferência (mecanismo de raciocínio)

- É um programa que utiliza a base de conhecimento como “dado” na solução de um problema.
- É um programa que utiliza mecanismos gerais de combinação de fatos e regras.

prof. Luis Otavio Alvares

- Base de conhecimento formada por fatos e por regras do tipo
SE condição ENTÃO conclusão
exemplos: se A e B então C
se G e H então I e J
- Quando o mecanismo de inferência é ativado, a base de conhecimento contém as informações representativas do enunciado do problema a resolver:
 - os fatos estabelecidos e os fatos a estabelecer (objetivos) formam a **base de fatos**
 - o conhecimento sobre o domínio constitui a **base de regras**

prof. Luis Otavio Alvares

Funcionamento

- O motor de inferência possui um raciocínio cíclico: um ciclo de base é composto por duas fases:
 - **avaliação**: procura das regras suscetíveis de serem ativadas em função do estado corrente da base de fatos e **escolha** das regras a ativar efetivamente
 - **execução**: modificação da base de fatos e eventualmente da base de regras
- Condição de parada:
 - encontrada solução para o problema
 - não há mais regra ativável

prof. Luis Otavio Alvares

Avaliação

- A fase de avaliação compreende 3 etapas:
 - seleção ou restrição
 - filtragem
 - resolução de conflitos

prof. Luis Otavio Alvares

Avaliação: seleção ou restrição

- Determina um subconjunto da base de regras e da base de fatos que, a priori, merece ser submetido à etapa seguinte de filtragem.
- exemplo:
 - 1- separar as regras já executadas
 - 2- separar fatos e regras em função do domínio:
 - distinguir entre regras para diagnóstico e regras para tratamento (área médica)
 - distinguir fatos e regras relativos a autoria de fatos e regras relativos a materialidade (área jurídica)
 - 3- separar fatos reconhecidos verdadeiros (estabelecidos, provados) dos fatos a estabelecer (objetivos, hipóteses).
Neste caso a restrição poderá consistir a privilegiar os fatos a estabelecer (hipóteses) em relação a outros, ou o problema mais recente em relação aos mais antigos

prof. Luis Otavio Alvares

Avaliação: filtragem

- É a segunda etapa da fase de avaliação
- Compara a parte condição de cada uma das regras selecionadas na etapa de restrição com o conjunto dos fatos considerados na mesma etapa. O subconjunto destas regras que têm condições de serem ativadas é chamado de **“conjunto de conflito”**.

prof. Luis Otavio Alvares

Avaliação: resolução de conflitos

- Na terceira etapa da fase de avaliação determina-se o subconjunto das regras que serão efetivamente ativadas.
- Normalmente esta escolha é baseada em critérios sem relação com o contexto da aplicação, como:
 - ordem de definição da base de regras (ex. Prolog)
 - ordem aleatória de execução
 - preferência para regras mais simples: menos condições ou menos variáveis a determinar antes da ativação

prof. Luis Otavio Alvares

Execução

- Constitui a segunda parte de cada ciclo
- O mecanismo de inferência comanda a ativação das regras selecionadas na fase de avaliação

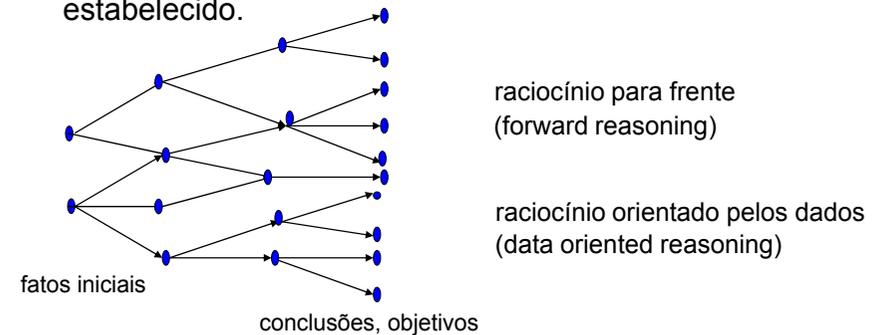
prof. Luis Otavio Alvares

Formas de trabalho do mecanismo de inferência

- Forma de encadeamento das regras
- Tipo de busca
- Monotônico x não monotônico

Forma de encadeamento das regras

- Encadeamento progressivo (forward chaining)
Os fatos da base de fatos sobre os quais é considerada a parte antecedente (condição, premissa) das regras representam informações cujo valor verdadeiro já foi estabelecido.



Exemplo: raciocínio progressivo

Base de regras:

Base de fatos: A,B

R1: se C então M
 R2: se A e D então E
 R3: se L então H
 R4: se B e C então G
 R5: se A e B então C
 R6: se G e D então H e I
 R7: se C então D
 R8: se E e K então H

Objetivo: H

Regra aplicada

R5
 R1
 R4
 R7
 R2
 R6

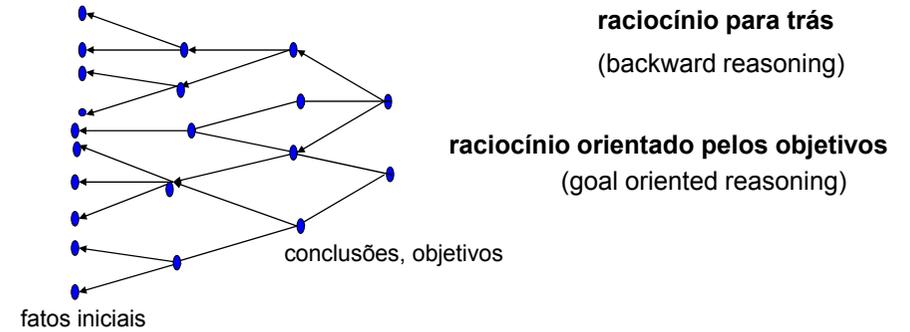
Base de fatos

A, B, C
 A, B, C, M
 A, B, C, M, G
 A, B, C, M, G, D
 A, B, C, M, G, D, E
 A, B, C, M, G, D, E, H, I

prof. Luis Otavio Alvares

Forma de encadeamento das regras

- Encadeamento regressivo (backward chaining)
 Parte-se dos objetivos e anda-se “para trás” através de subobjetivos



prof. Luis Otavio Alvares

Exemplo: raciocínio regressivo

Base de regras:

Base de fatos: A,B

R1: se C então M
 R2: se A e D então E
 R3: se L então H
 R4: se B e C então G
 R5: se A e B então C
 R6: se G e D então H e I
 R7: se C então D
 R8: se E e K então H

Objetivo: H

Regra

Metas

Base de fatos

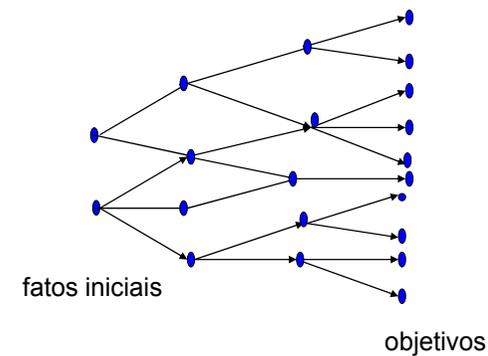
Regra	Metas	Base de fatos
	H	A, B
R3	L	A, B, H
	H	A, B
R6	G, D	A, B, H, I
R4	D, C	A, B, H, I, G
R7	C	A, B, H, I, G, D
R5	--	A, B, H, I, G, D, C

IMPASSE

prof. Luis Otavio Alvares

Tipo de busca

em profundidade x em amplitude



prof. Luis Otavio Alvares

Raciocínio monotônico x não monotônico

Raciocínio monotônico

- nenhum fato ou regra pode ser retirado da base de conhecimento durante uma seção de consulta
- nenhum fato novo fará com que um “fato antigo” torne-se inválido ou não-verdadeiro

Vantagens

- quando um fato novo for acrescentado (deduzido), nenhuma conferência precisa ser feita para ver se há inconsistência entre o fato novo e a base antiga
- não é necessário guardar, para cada novo fato, a lista dos fatos que possibilitaram a dedução do novo fato, pois não há perigo de os fatos antigos deixarem de ser verdadeiros

prof. Luis Otavio Alvares

Raciocínio monotônico

• Desvantagens - nas situações de:

- informação incompleta (raciocínio por omissão)
- situação em mudança
- geração de suposições no processo de resolver problemas complexos como planejamento

prof. Luis Otavio Alvares

Exemplo de algoritmo para motor de inferência com encadeamento progressivo

prof. Luis Otavio Alvares

```
procedure ESTABELECEER-UM-FATO (FATO)
  if FATO ∈ BASE-DE-FATOS
    then exit “fato provado”
  exit EXECUTAR-UM-CICLO (BASE-DE-REGRAS)

procedure EXECUTAR-UM-CICLO (AS-REGRAS)
  if AS-REGRAS = ∅
    then exit “fato não provado”
  UMA-REGRA ← uma regra de AS-REGRAS, por exemplo a primeira
  AS-REGRAS ← AS-REGRAS exceto UMA-REGRA
  if todos os fatos da condição de UMA-REGRA ∈ BASE-DE-FATOS
    then begin
      if a conclusão de UMA-REGRA é FATO
        then exit “fato provado”
      adicionar a conclusão de UMA-REGRA à BASE-DE-FATOS
      BASE-DE-REGRAS ← BASE-DE-REGRAS exceto UMA-REGRA
      exit EXECUTAR-UM-CICLO (BASE-DE-REGRAS)
    end
  exit EXECUTAR-UM-CICLO (AS-REGRAS)
```

prof. Luis Otavio Alvares

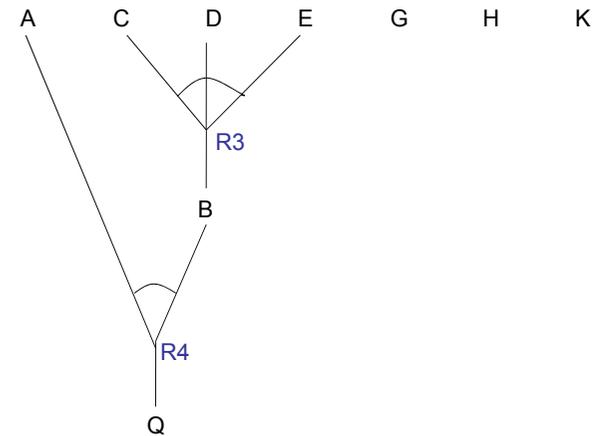
Base de regras:
 R1: se K e L e M então I
 R2: se I e L e J então Q
 R3: se C e D e E então B
 R4: se A e B então Q
 R5: se L e N e O e P então Q
 R6: se C e H então R
 R7: se R e J e M então S
 R8: se F e H então B
 R9: se G então F

Fato a obter: Q

Basedefatos: A,C,D,E,G,H,K

prof. Luis Otavio Alvares

Grafo e/ou para o algoritmo



prof. Luis Otavio Alvares

Características do algoritmo

- encadeamento progressivo
- o novo fato gerado pela aplicação de uma regra é imediatamente considerado pelas próximas regras a serem executadas (em profundidade)
- a ordem das regras é importante para o tempo de execução, mas não para o resultado

prof. Luis Otavio Alvares

procedure ESTABELEECER-UM-FATO (FATO)

if FATO ∈ BASE-DE-FATOS

then exit "fato provado"

FATOSNOVOS ← lista vazia

exit EXECUTAR-UM-CICLO (BASEDEREGRAS, FATOSNOVOS)

procedure EXECUTAR-UM-CICLO (ASREGRAS, FATOSNOVOS)

if ASREGRAS é vazio

then begin

if FATOSNOVOS é vazio

then exit "insucesso"

BASEDEFATOS ← BASEDEFATOS acrescida de FATOSNOVOS

FATOSNOVOS ← lista vazia

end

UMAREGRA ← uma regra de ASREGRAS, por exemplo a primeira

ASREGRAS ← ASREGRAS diminuída de UMAREGRA

if todos os fatos da premissa de UMAREGRA pertencem à BASEDEFATOS

then begin

if a conclusão de UMAREGRA é o FATO

then exit "fato provado"

acrescentar a conclusão de UMAREGRA à FATOSNOVOS (se a conclusão já não pertence a BASEDEFATOS ou a FATOSNOVOS)

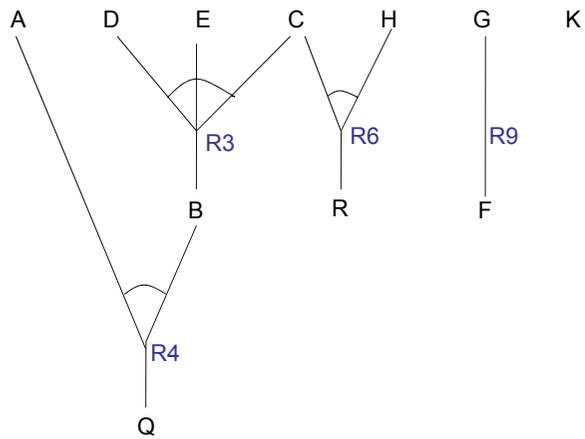
BASEDEREGRAS ← BASEDEREGRAS diminuída de UMAREGRA

end

exit EXECUTAR-UM-CICLO (ASREGRAS, FATOSNOVOS)

prof. Luis Otavio Alvares

Grafo e/ou do algoritmo



prof. Luis Otavio Alvares

Características do algoritmo:

- encadeamento progressivo
- o **novo fato** gerado pela aplicação de uma regra **não** é imediatamente considerado pelas próximas regras a serem executadas (em amplitude)
- a ordem das regras **não** é importante para o tempo de execução, nem para o resultado

prof. Luis Otavio Alvares