Encadeamento progressivo: Exemplo no domínio dos veículos

- Carregar a BR de veículos no MI e atribuir valores iniciais para algumas variáveis, guardando esses fatos na MT.
  - Fatos iniciais: num-rodas=4, motor=sim, num-portas=3, tamanho=médio
- Fase de “casamento”
  - Conjunto de conflito da 1ª rodada de inferência resulta em apenas uma regra

Automóvel: Se num-rodas=4

E motor=sim

Então veículoTipo=automóvel

Encadeamento progressivo: Exemplo no domínio dos veículos

- A resolução de conflito fica então trivial.
- Fatos na MT:
  - num-rodas=4; motor=sim; num-portas=3; tamanho=médio
  - *veículoTipo=automóvel*
- Casamento: **segunda rodada de inferência seleciona apenas 1 regra para o conjunto de conflito:**
  - **MiniVan: Se** veículoTipo=automóvel

    E tamanho=médio

    E num-portas=3

    Então *veículo=MiniVan*

Encadeamento progressivo: Exemplo no domínio dos veículos

- Fatos na MT:
  - num-rodas=4; motor=sim; num-portas=3; tamanho=médio
  - veículoTipo=automóvel; *veículo=MiniVan*
- Casamento:
  - terceira rodada de inferência seleciona a mesma regra que na rodada anterior
  - como esta já foi disparada, não será adicionada novamente ao conjunto de conflito
  - com o conjunto de conflito vazio, o processo de inferência para
- Com os fatos na MT, concluímos então que o veículo procurado é uma *Minivan*.

Exemplo: regras disparadas

- O fluxo de informações se dá através de uma série de regras encadeadas a partir das premissas para as conclusões

Automóvel: Se num-rodas=4

E motor=sim

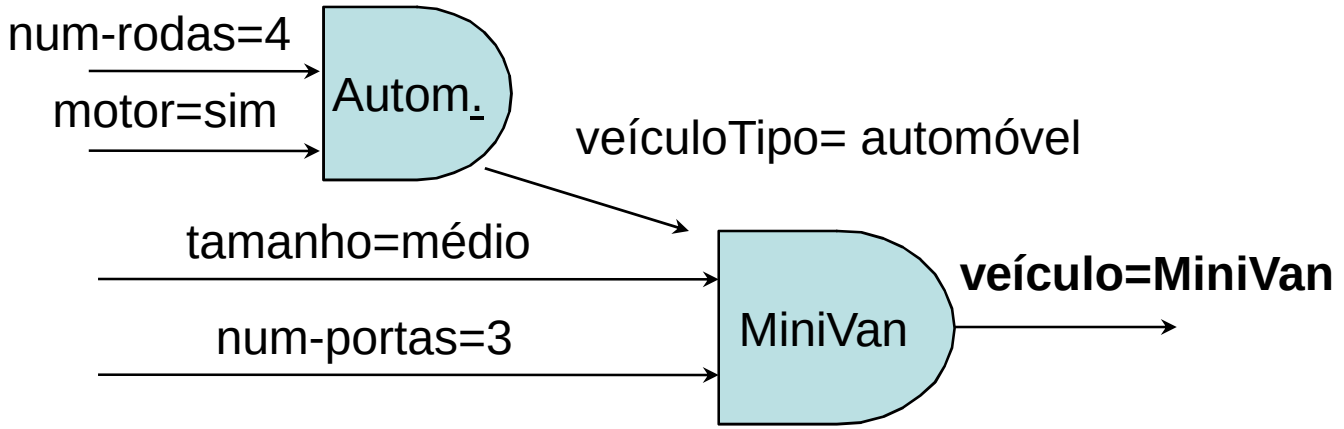
Então *veículoTipo=automóvel*

MiniVan: Se veículoTipo=automóvel

E tamanho=médio

E num-portas=3

Então *veículo=MiniVan*



Encadeamento regressivo: Busca e Casamento

- Da hipótese aos dados
  - Parte da hipótese que se quer provar, procurando regras na BR cujo conseqüente satisfaz essa hipótese.
  - usa as regras da BR para responder a perguntas
  - busca provar se uma asserção é verdadeira
    - ex.: criminoso(Fulano)?
  - só processa as regras relevantes para a pergunta
- Duas etapas:
  - Busca e Casamento (unificação)
- Utilizado em sistemas de aconselhamento
  - trava um “diálogo” com o usuário
  - ex.: MYCIN

Encadeamento regressivo: algoritmo

1. Armazena as regras da BC na máquina de inferência (MI) e os fatos na memória de trabalho (MT);
2. Adiciona os dados iniciais à memória de trabalho;
3. Especifica uma variável objetivo para a MI;
4. Busca o conjunto de regras que possuem a variável objetivo no conseqüente da regra

   (Isto é, seleciona todas as regras que atribuem um valor à variável objetivo quando disparadas.)

   Insere as regras selecionadas na pilha de objetivos;

5. Seleciona a regra no topo da pilha de objetivos

- Se a pilha de objetivos está vazia, o algoritmo falha!

(não conseguiu provar a hipótese de entrada)

6. Tenta provar que a regra selecionada é verdadeira testando, um a um, se todos os seus antecedentes são verdadeiros:

a) se o 1o. antecedente é V, vá em frente para o próximo

b) se algum antecedente dessa regra for F, a regra toda falha;

- o algoritmo volta ao passo 5 (para tentar provar outra regra selecionada previamente, disponível na pilha de objetivos)

6. Continuação:

c) quando todos os antecedentes são provados V

- dispara a regra = instancia a variável no seu conseqüente para o valor que aparece nessa regra e

- devolve o resultado para o usuário (o algoritmo termina com sucesso).

d) se o valor-verdade de um antecedente é desconhecido (porque não está na MT):

- suspende o processamento da regra atual

- vai para o passo 4 com essa variável como variável objetivo.

(nesse caso, o algoritmo cria uma nova pilha de objetivos, com base na nova variável objetivo – RECURSÃO!)

6. Continuação:

d) continuação:

- Se conseguir provar que o valor-verdade dessa nova variável é V:

    - dispara a regra, instancia a variável no seu conseqüente para o valor que aparece nessa regra;

    - abandona a nova pilha de objetivos e

    - retoma o processamento da regra que estava sendo provada antes (6.a)

- Se o valor-verdade dessa nova variável é F:

    - abandona a regra e volta para a nova pilha de objetivos

    - se nova pilha de objetivos estiver vazia, o algoritmo falha.

6. Continuação:

d) continuação:

- Se o valor-verdade de um antecedente dessa nova regra sendo testada é desconhecido

- suspende o processamento da regra atual

- vai para o passo 4 com essa variável como variável objetivo.

(RECURSÃO de novo!)

Encadeamento regressivo: Busca e Casamento

- O sistema percorre a BC em busca regras cujo conseqüente “casa” com a hipótese de entrada
  - Unificação é realizada com busca em profundidade
- Se a hipótese de entrada é um fato (ex. criminoso(Fulano)),
  - a busca para quando encontra a 1ª regra que casa com o fato
  - o sistema devolve uma variável booleana (V ou F).

Encadeamento regressivo: Busca e Casamento

- Se a hipótese tem alguma variável livre (ex. criminoso(X)),
  - o sistema (programador) pode optar por devolver a 1ª instanciação encontrada, ou
  - devolver uma lista com todas as possíveis instanciações para aquela variável.
- Portanto, não há conflito de execução de regras!

Encadeamento regressivo: Exemplo no domínio dos veículos

- Carregar a BR de veículos na MI e os fatos na MT
- Fatos iniciais:
  - num-rodas=4, motor=sim, num-portas=3, tamanho=médio
- Especificar variável objetivo
  - veículo=?
- Pilha de objetivos
  - regras com variável objetivo no conseqüente
    - as 7 primeiras regras da nossa BC

Encadeamento regressivo: Exemplo no domínio dos veículos

- Tenta provar verdadeiros os antecedentes da 1ª regra usando busca em profundidade
  - **Bicicleta: Se** veículoTipo=ciclo
    E num-rodas=2
    E motor=não
    Então veículo=*Bicicleta*
- *VeículoTipo=ciclo* não aparece na MT
  - nova variável objetivo
- Atualiza pilha de objetivos
  - inclui regras com nova variável objetivo no conseqüente
    - apenas a penúltima regra da nossa BC

Encadeamento regressivo

- *veículoTipo=ciclo* só é verdade em apenas uma regra
  - **Ciclo: Se** num-rodas < 4

    Então veículoTipo=*ciclo*

- Verifica o valor verdade dos antecedentes da regra
  - num-rodas < 4 ===> FALSO!

- Donde se deduz que *veículo=Bicicleta* é Falso!

Encadeamento regressivo

- Desempilha as outras regras, uma a uma, até encontrar a regra abaixo - que vai dar certo!
  - **MiniVan: Se** veículoTipo=automóvel
    E tamanho=médio
    E num-portas=3
    Então veículo=*MiniVan*
- *VeículoTipo=automóvel* não existe na MT
  - **Automóvel: Se** num-rodas=4 OK! (1)
    E motor=sim OK! (2)
    Então veículoTipo=*automóvel* ===> *OK! (3)*
- Tenta provar os outros antecedentes da regra, que estão todos instanciados na MT, e são verdadeiros!
- veículo=*MiniVan* é verdade!

Encadeamento regressivo

- Se o fato a ser provado não aparece explicitamente na base e nem pode ser deduzido por nenhuma outra regra, duas coisas podem ocorrer, dependendo da implementação do sistema
  - o fato é considerado FALSO
    - ex. Prolog
  - o sistema consulta o usuário via sua interface
    - ex. Sistema ExpertSinta

Regras com fator de incerteza

- Geralmente, é necessário associar-se um fator de incerteza (ou de confiança) a algumas regras na BR
- Incerteza nos dados e na aplicação das regras

  If (previsão-do-tempo = chuva) > 80%

  and (previsão-períodos-anteriores = chuva) = 85%

  then (chance-de-chuva = alta) = 90%

- Infelizmente ...
  - combinar as incertezas dos antecedentes neste caso propaga erros
  - só uma abordagem probabilista pode tratar este tipo de incerteza corretamente

Exemplos de SE

Histórico: GPS (1960s)

- General Problem Solver (GPS)
- Motivação:
  - leis do pensamento + máquinas poderosas
- Funcionamento:
  - ≅ planejamento + sub-goaling
    - ex. estou com fome => comer => pedir pizza => telefonar => ir para a sala => sair do quarto...
- O Logic theorist deu certo mas.... em geral, GPS não funciona
  - fraca representação de conhecimento
  - humanos são bons só em domínios restritos

Histórico: Primeiros SEs (1960s-1970s)

- DENDRAL
  - Inferir estrutura molecular de componentes desconhecidos dadas a massa espectral e a resposta nuclear magnética
  - Conhecimento especializado poda a busca por possíveis estruturas moleculares
  - Fez sucesso: publicações científicas
  - Representação procedimental de conhecimento

Histórico: Primeiros SEs (1960s-1970s)

- MYCIN
  - Diagnosticar rapidamente meningite e outras infecções bacterianas, e prescrever tratamento
  - Representação de conhecimento baseada em regras probabilísticas (em torno de 500)
  - Fez sucesso: acima de 90% de acerto
  - introduziu explicação e boa interface com usuário
- Exemplo de regra

```
ifthe infection is meningitis and
   the     type of infection is bacterial and
   the patient has undergone surgery and
   the patient has under gone  neurosurgery and
   the neurosurgery-time was < 2 months ago and
   the patient got a ventricular-urethral-shunt
then  infection = e.coli(.8)  or klebsiella(.75)
```

Histórico: 1970s & 1980s

- 1970s: Esforço para desenvolver melhores (e mais especializadas)
  - Linguagens de representação de conhecimento
  - Mecanismos de inferência
- Conclusões
  - O poder de um sistema é derivado do conhecimento específico que ele possui, e não de esquemas de inferências e formalismo particular que ele emprega
  - As linguagens existentes já bastam
- 1980s: Grande *boom* dos SEs
  - XCON, XSEL, CATS-1, etc.

CATS-1

- Problema da General Electric:
  - Aposentadoria de David Smith: engenheiro especialista em falhas de motores elétrico-diesel de locomotivas
  - Custo deste tipo de engenheiro
- Solução convencional
  - Treinamento de engenheiros novatos
- 1980: Construção de CATS-1 (DELTA)
  - Meses de entrevista, 3 anos p/ primeiro protótipo
  - Permite diagnostico em poucos minutos
  - Existe um em cada oficina
  - Dá treinamento: é amigável e explica decisões

Balanço

Benefícios do S.E.

- Criação de repositório de conhecimento
- Crescimento de produtividade e qualidade
- Habilidade de resolver problemas complexos
- Flexibilidade e modularidade
- Operação em ambientes arriscados
- Credibilidade
- Habilidade de trabalhar com informações incompletas ou incertas
- Fornecimento de treinamento

Problemas e Limitações

- Avaliação de desempenho difícil
- É difícil extrair conhecimento especialista
- Só trabalham muito bem em domínios estreitos
- Engenheiros de Conhecimento são raros e caros
- Transferência de conhecimento está sujeito a um grande número de preconceitos

Últimos desenvolvimentos

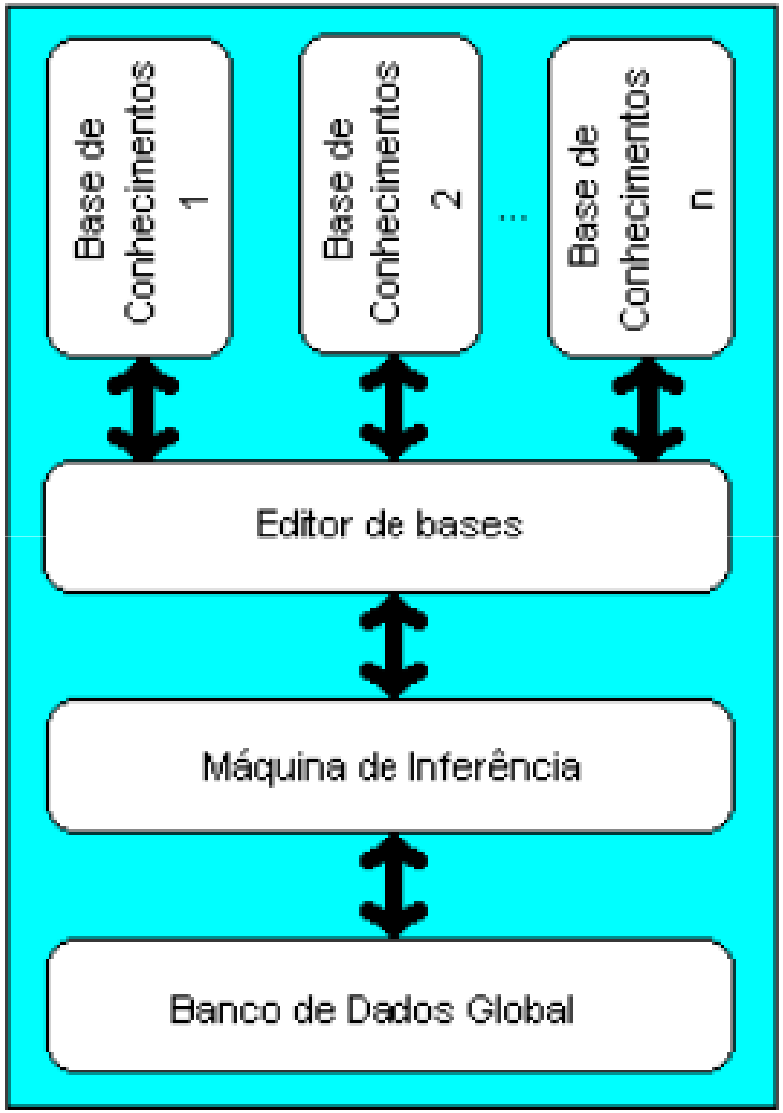
- Aquisição de conhecimento:
  - SEs de 2ª geração & aprendizagem
- Integração com outros sistemas
  - ex. banco de dados e sistemas de suporte à decisão
- Tratamento de incerteza

Expert SINTA

Expert SINTA

- Desenvolvido no laboratório de Inteligência Artificial da Universidade Federal do Ceará (UFC)
- Uma ferramenta visual para criação de sistemas especialistas
- Utiliza regras de produção (SE… Então…)
- Ferramenta *shell*
  - Diversos sistemas podem compartilhar uma mesma máquina de inferência e outras caracteristicas comuns;
  - Focam o implementador em apenas representar o conhecimento do especialista;
  - Fornece suporte para a construção de interfaces
  - Possui tratamento probabilistico
  - Utiliza raciocínio regressívo
  - Utiliza fatores de confiança
  - Disponibiliza ferramentas de depuração

Sistemas Especialistas



Arquitetura simplifica do Expert SINTA

Expert SINTA

- Regras de produção
  - Modularidade;
  - Facilidade de edição (manutenibilidade);
  - Transparência do sistema.
- Regras no Expert SINTA

  Se esposa autoriza ou vou escondido

  E dinheiro>= valor ingresso

  ENTÃO vou para o esfrega [90%]

Denomina-se os conseqüentes de uma regra como as **cabeças** da regra e os antecedentes, **caudas**. (proveniente de PROLOG)

Expert SINTA

Critérios para definições de assertivas no Expert SINTA :

(I). A estrutura de cada cauda (premissa) deve obedecer ao seguinte modelo:

<conectivo> <atributo> <operador> <valor>

<conectivo>: NÃO, E, OU

<atributo>: variável capaz de assumir uma ou múltiplas instanciações no decorrer da consulta à base de conhecimentos

<operador>: =, >, <=, <>,...

<valor>: item de uma lista a qual foi previamente criada e relacionada a um atributo.

Expert SINTA

(II). A estrutura de cada cabeça (conclusão) deve obedecer ao seguinte modelo:

<atributo> = <valor> <grau de confiança>

<atributo>: equivalente ao usado em caudas

“=”: operador de atribuição e não de igualdade

<valor>: equivalente ao usado em caudas

<grau de confiança>: porcentagem indicando a confiabilidade daquela conclusão específica da regra. O grau de confiança varia de 0% a 100%

Expert SINTA

O raciocínio regressivo (encadeamento para trás) destaca-se em problemas nos quais há um grande número de conclusões que podem ser atingidas, mas o número de meios pelos quais elas podem ser alcançadas não é grande, e em problemas nos quais não se pode reunir um número aceitável de fatos antes de iniciar-se a busca por respostas.

O encadeamento para trás também é mais intuitivo para o desenvolvedor, pois é fundamentada na recursão, um meio elegante e racional de programação, para onde a própria Programação em Lógica se direcionou. Em nenhum momento, porém, deixa-se de reconhecer que o encadeamento para frente possui vantagens em determinadas ocasiões.

UNIVASF

Exemplo de encadeamento para trás

Sejam as seguintes regras um sistema especialista para “decidir se devo ou não ir à praia amanhã”.

- **REGRA 1**
  SE amanhã pode chover = Não
  E tenho dinheiro suficiente = Sim
  E tenho tempo suficiente = Sim
  ENTÃO devo ir à praia = Sim
- **REGRA 2**
  SE amanhã pode chover = Sim
  OU tenho dinheiro suficiente = Não
  OU tenho tempo suficiente = Não
  ENTÃO devo ir à praia = Não
- **REGRA 3**
  SE o serviço de meteorologia disse que vai chover amanhã = Sim
  ENTÃO amanhã pode chover = Não
- **REGRA 4**
  SE não vou sair hoje = Sim
  E nenhuma emergência ocorrer = Sim
  ENTÃO tenho dinheiro suficiente = Sim
- **REGRA 5**
  SE minha namorada ligar = Sim
  ENTÃO não vou sair hoje = Não
- **REGRA 6**
  SE meu orientador passar trabalho extra = Sim
  ENTÃO tenho tempo suficiente = Não

Expert SINTA

Examinando o SE percebe-se que o objetivo é determinar o valor da variável “devo ir à praia”.

O Expert SINTA procura as regras nas quais a variável sendo procurada no momento pode receber um valor se a regra for aceita (ou seja, quando ela aparece após o ENTÃO).

Em seguida, a máquina de inferência verifica se a regra vale ou não.

No exemplo dado, o sistema avalia a regra 1 para poder determinar se devo ir à praia ou não.

Mas, para isso, temos que determinar se todas as premissas (amanhã pode chover = Não, tenho dinheiro suficiente = Sim, tenho tempo suficiente = Sim) são verdadeiras.

Expert SINTA

Para descobrir se amanhã pode chover, tenho que repetir o processo, avaliando a regra 3. Agora tenho que saber se a meteorologia disse sobre a possibilidade de chuva amanhã.

Como não existe nenhuma regra que possa concluir o que a meteorologia disse, o Expert SINTA realiza uma pergunta ao usuário, do tipo “A meteorologia afirmou se amanhã pode chover (Sim/Não)?” (claro, o computador não irá criar uma frase, essa é uma das tarefas do criador da base de conhecimento).

Se a meteorologia disser que amanhã vai chover, então primeira premissa da regra 1 passará no teste (quem confia nos serviços meteorológicos?).

Expert SINTA

Avaliando a segunda premissa e repetindo o processo do encadeamento para trás, chegamos à regra 4. Mas, para descobrir se eu vou sair hoje ou não, é preciso recorrer à regra 5.

A regra 5 leva a uma pergunta, pela qual concluiremos se tal regra vai ser aprovada. Se realmente for aprovada, a primeira premissa da regra 4 é verdadeira, fazendo com que verifiquemos agora a segunda premissa.

Quando todas as premissas da primeira regra forem avaliadas, podemos determinar se a regra 1 foi aprovada ou não. Caso contrário, passamos para a próxima regra que possa concluir um valor para a variável devo ir à praia.

Expert SINTA

Variáveis univaloradas x variáveis multivaloradas

- Uma única variável pode receber vários valores em uma única consulta ao sistema. É muito comum, por exemplo, em sistemas de diagnóstico médico, onde o paciente pode apresentar mais de uma doença. Portanto, é importante saber lidar com variáveis que podem ter apenas uma instanciação (univalorada) ou múltiplas (multivaloradas).
- Quando a máquina de inferência está atrás de encontrar instanciações para uma variável univalorada, ela irá procurar até encontrar um valor ou até esgotar todas as possibilidades da base de conhecimento. Se, por algum motivo, durante a busca de uma outra variável, uma variável univalorada receber um valor quando já possuía outro, esse valor antigo será descartado e o novo vigorará.

Expert SINTA

Variáveis univaloradas x variáveis multivaloradas

- A busca de valores para variáveis multivaloradas prossegue até que toda a base de conhecimento seja explorada. Os valores permanecem acumulados. É nessa hora que é preciso ter cuidado com contradições presentes na base. O Expert SINTA, na presente versão, não faz verificações de inconsistências lógicas.

Obs.: variáveis numéricas são tratadas como univaloradas, sempre.

Expert SINTA

Conhecimento monotônico x conhecimento não-monotônico

No dia-a-dia, enfrentamos situações que modificam as nossas certezas. O que antes tínhamos por certo agora estamos convictos que não são mais realidade. Isso porque nós pensamos em um tipo de lógica não-monotônica, ou seja, podemos receber informações que contrarie as que já possuímos.

No tratamento tradicional de informações da Inteligência Artificial, o conhecimento monotônico, ou seja, aquele que ao acrescentarmos novas informações nunca fica contraditório, inconsistente, é o comumente usado. O Expert SINTA trata o conhecimento de forma essencialmente monotônica.