Sistemas Baseados em Conhecimento e Sistemas Especialistas

Inteligência Artificial

Sistemas Baseados em Conhecimento

- SBCs são sistemas que usam:
  - conhecimento especializado;
  - representado explicitamente ;
  - para resolver problemas complexos.
- São sistemas capazes de...
  - Questionar o usuário;
  - Raciocinar com base no conhecimento disponível;
  - Explicar seu raciocínio ao usuário;
  - "Lidar" com seus erros.

Sistemas Baseados em Conhecimento

- Pode ser usado para:
  - Guiar a seleção, localização e uso de regras;
  - Dar informação acerca das regras e do conhecimento;
  - Justificar as regras melhorando as capacidades de explicação;
  - Apoiar na detecção de erros ao introduzir novas regras;
  - Facilitar a introdução de novo conhecimento.

Relembrando... Sistemas BC *X* Agentes BC

- Sistemas baseados em conhecimento:
  - Têm uma base de conhecimento e uma máquina de inferência associadas;
  - Formalizam e implementam parte dos agentes BC.
- Qual a diferença?
  - Agentes interagem com o ambiente onde estão imersos através dos sensores e atuadores.

Sistemas Baseado em Conhecimento (SBC)

- Ou **Knowledge-Based Systems** (KBS);
- Principais diferenças entre um SBC e os convencionais:
  - Organização dos dados;
  - SBCs: métodos que fazem busca em um espaço de possíveis soluções, e fazem uso intensivo de heurísticas para tornar a busca mais efetiva
    - Sistema Convencional (SC): Algoritmos determinísticos para realizar suas funções!
  - Separação do conhecimento e método de solução:
    - Maior capacidade de explicação

Comparação SC versus SE

Sistema Convencional	Sistema Especialista
Numérico	Simbólico
Algorítmico	Heurístico
Informação precisa	Informação imprecisa
Interface por comandos	Diálogo natural com explicação
Proporciona solução final	Recomendação com explicação
Solução ótima	Solução aceitável

Sistemas Especialistas

- Sistema Especialista (SE)?
  - O que é um especialista?
  - O que é *expertise* (competência)?
- Qual é a estrutura de um SE?
- Quem usa SE?
- Como ele pode ser usado?
- Quais são os potenciais benefícios?
- Quais são as possíveis limitações?

O que é um Sistema Especialista?

- ***Sistema:***
  "Conjunto de elementos, materiais ou ideais, entre os quais se possa encontrar ou definir alguma relação"
- ***Especialista***
  "Pessoa que se consagra com particular interesse e cuidado a certo estudo. Conhecedor, perito"
- **Sistemas Especialistas**
  "São sistemas que solucionam problemas que são resolvidos apenas por pessoas especialistas (que acumularam conhecimento exigido) na resolução destes problemas"

O que é um Sistema Especialista?

- Ramo da IA:
  - Faz uso intensivo do conhecimento especializado para resolver problemas ao nível de um especialista humano;
  - Programas computacionais que emulam o comportamento de especialistas humanos em algum domínio específico do conhecimento.

Sistemas Especialistas

- Sistemas Especialistas (SE) são aplicações que têm por objetivo resolver problemas complexos de forma idêntica à utilizada pelos peritos humanos;
- Os SEs são um caso específico de Sistemas Baseados em Conhecimento
  - Num SE o conhecimento é obtido a partir de um ou mais peritos ou especialistas.
- O desenvolvimento de um SE incorpora, para além de uma vertente técnica, uma vertente humana complexa
  - Relacionamento de confiança que se estabelece entre quem especifica e desenvolve o sistema e quem possui o conhecimento.

Um Especialista, por definição ...

- Identifica questões relevantes ao problema;
- Resolve problemas complexos rapidamente;
- Explica o resultado;
- Aprende continuamente (reestrutura o conhecimento);
- Sabe quando aplicar "exceções";
- É humano!

O que é *Expertise* (competência)?

- Conhecimento profundo de uma tarefa específica, adquirido através de treinamento, leitura, experiência etc.
- O que é conhecimento?
  - Dados + processamento = informação
  - Informação + processamento (experiência, treinamento etc.) = **conhecimento**

Um Especialista também é...

- Caro;
- Raro;
- Ocupado;
- Inconsistente;
- Emocional;
- Mortal.

Todas estas são boas razões para considerarmos a captura de sua competência.

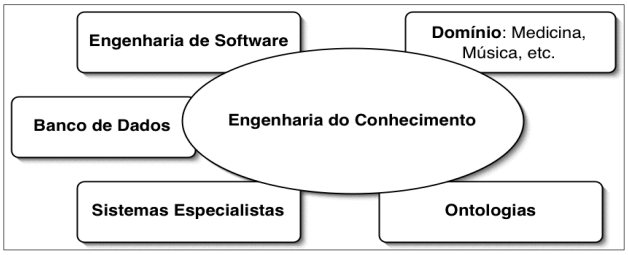
Características dos Sistemas Especialistas

- Faz inferências e deduções a partir de informações fornecidas pelo usuário;
- O **conhecimento** é aplicado na solução do problema, usado para guiar e restringir a busca por soluções;
- A área do problema é pequena e bem definida.

O que é um Sistema Especialista?

- Um Sistema Especialista é aquele que é:
  - Projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano;
  - Capaz de emitir uma decisão, apoiado em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano;
  - Além de inferir conclusões, deve ter capacidade de aprender novos conhecimentos
    - Melhorando o seu desempenho de raciocínio, e a qualidade de suas decisões.

Engenharia do Conhecimento - Uma Área Multidisciplinar!



Engenharia do Conhecimento

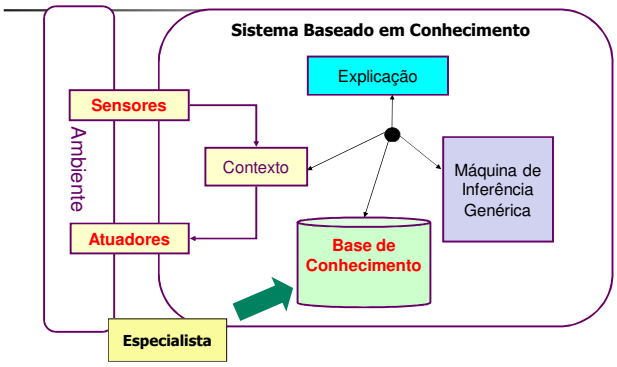
- Engenharia do Conhecimento
  - estuda *como* construir uma **Base de Conhecimento** (BC)

1. Nível do conhecimento: aquisição de conhecimento
   - Conhecimento em "estado puro" - linguagem natural
     - ex. táxi automático: a ponte Princesa Isabel liga a Rua da Imperatriz à Rua Nova
2. Nível lógico: formalização
   - Conhecimento codificado em sentenças - linguagem formal
     - ex. sentença lógica: liga(Ponte-PI,RI,RN)
3. Nível de máquina: implementação
   - Estrutura de dados representando as sentenças do nível lógico
     - ex. listas, tabelas, objetos, etc.

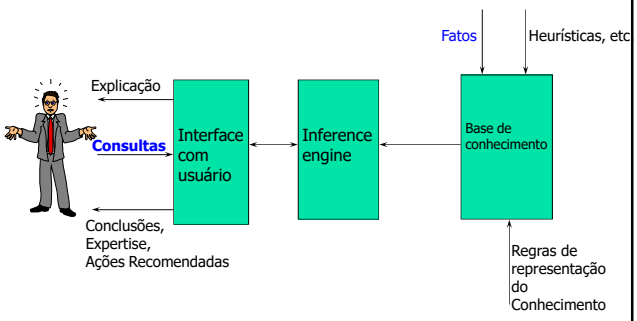
EC - Conceitos Básicos

- Engenheiro de conhecimento (EC)
  - Guia a aquisição, a criação da representação do conhecimento especializado, a implementação e o refinamento do SBC.
- *Expertise*
  - Conhecimento especializado adquirido por longo treinamento, leitura e experiência.
- Especialista (*Expert*)
  - Quem possui conhecimento especializado, experiência e métodos, e a habilidade de aplicá-los para dar "conselhos" e resolver problemas

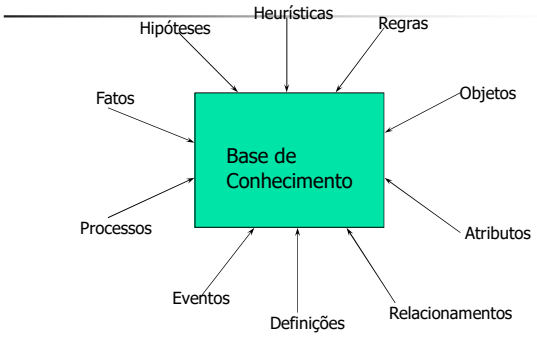
Qual a principal diferença entre um sistema especialista e um SBC?



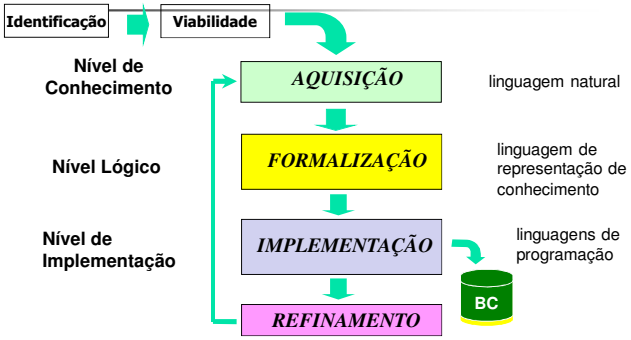
Arquitetura de SBC



Base de conhecimento

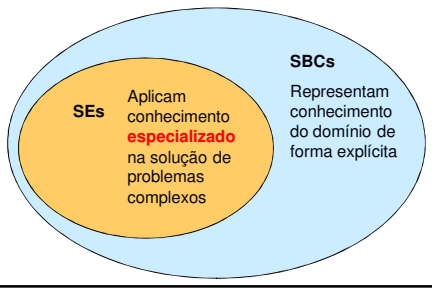


Ciclo de vida dos sistemas baseados em conhecimento



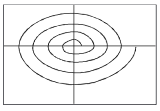
Engenharia do Conhecimento lida com dois tipos de sistemas

- SEs = Sistemas Especialistas



Etapas do desenvolvimento de SBCs

1. Construção da base de conhecimento:
   - Aquisição de conhecimento;
   - Representação do conhecimento (formalização)
2. Implementação do sistema:
   - Codificação;
   - Construção do sistema de explicação, interface, etc.
3. Refinamento e validação



Etapas do desenvolvimento de SBCs
mais detalhes...

1. Planejamento do sistema:
   - Identificação do Domínio;
   - Seleção da equipe;
   - Seleção da ferramenta de desenvolvimento.
2. Aquisição do conhecimento:
   - Identificação do conhecimento a adquirir;
   - Conceituação;
   - Formalização da BC.

Etapas do Desenvolvimento de SBCs
mais detalhes...

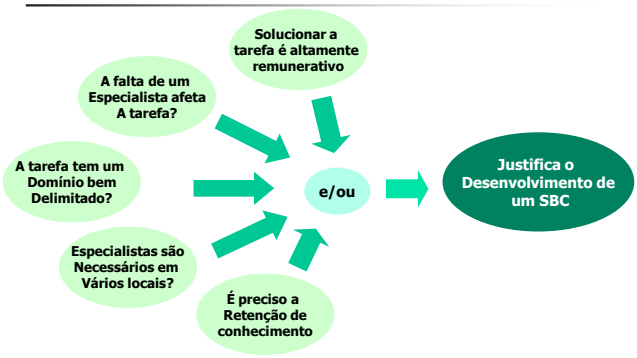
3. Implementação:
   - Criação de uma representação do conhecimento;
   - Implementação da Interface;
   - Documentação.
4. Validação e Refinamento:
   - Validação;
   - Refinamento.

E tudo começa novamente...

Ciclo de vida dos sistemas baseados em conhecimento

- Identificação do domínio:
  - Descreve o domínio de conhecimento, termos chaves e referências;
  - Resumo simplificado dos conceitos relacionados ao domínio;
  - Análise funcional: entradas e saídas identificadas.

Análise de Viabilidade



Aquisição de conhecimento

- Primeira e mais importante fase do ciclo de vida de um SBC;
- Conhecimento é adquirido (especialista, livros, etc.)
  - Acompanha toda a vida útil do sistema
- Passos:
  - Identificação;
  - Conceituação;
  - Formalização;
  - Implementação.

Aquisição de conhecimento

- Conceituação: trabalha diretamente com o conhecimento do especialista;
- Interação com o especialista, tarefa difícil:
  - Diversos tipos e níveis de conhecimento;
  - Verbalização: difícil aos humanos (conhecimento implícito);
  - Conhecimento especializado: rico e complexo;
  - Especialista: fornecer detalhes do conhecimento;
  - Problemas com a linguagem;
  - Trabalho com mais de um especialista.

Aquisição de conhecimento

- Aquisição automática de conhecimento:
  - Suavizar o problema da expressão verbal;
  - Criar sistemas capazes de atualizar, refinar e acrescentar conhecimento;
  - Interagir com o especialista, visando o aprendizado do SE:
    - Automática (KADs -*Knowledge* Aided Design)
    - Semi-automática (editores de protocolos, gráficos, etc.)

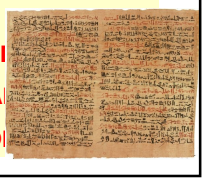
Aquisição de Conhecimento (AC)

- Conhecimento descrito através das linguagens:
  - Natural;
  - Diagramática;
  - Semi-formais;
  - Formais.
- Métodos e técnicas para se utilizar durante a AC → adquirir o máximo possível de conhecimento
- Técnicas
  - Entrevistas
    - Não estruturada;
    - Estruturada.
  - Observações
    - Simples;
    - Análise de protocolo.
  - Análise por interrupção
    - Informação limitada;
    - Processamento limitado.

O Papiro de Edwin Smith

- O papiro de Edwin Smith é um texto egípcio de 3700-anos de idade, encontrado em Luxor.

ABCDEECDBBACDACDBCDECDADCADBADE
ECDBBACDACDBCDECDADCADBADCDBBACDA
BCDEECDBBACDACDBCDECDAD
BBACDACDBCDECDADCADBADEDCI
DCDBBADCDBBABCDECDADCADBA
BACDACDBCDECDADBACDACDBCD



O Papiro de Edwin Smith

- Ele contém a descrição médica de 48 tipos diferentes de ferimentos na cabeça;
- O papiro utiliza um formato fixo para descrição dos problemas, composto de: título – sintomas – diagnósticos – prognóstico e tratamento.

O Papiro de Edwin Smith

- Há um estilo fixo para cada uma das partes de descrição de problema.
- Desta forma o prognóstico sempre terá o formato: "É um ferimento que consigo curar", ou "É um ferimento que irei combater", ou "É um ferimento que nada posso fazer".
- Um exemplo extraído do papiro de Edwin Smith:

O Papiro de Edwin Smith

Titulo:

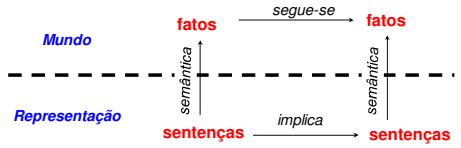
Instruções para tratamento dos seios da face.

Sintomas:

No exame de um homem com fratura dos seios da face, deve-se encontrar uma saliência, e uma mancha vermelha nas bordas do ferimento.

Representação do Conhecimento

- Representar o conhecimento adquirido do especialista num formalismo passível de entendimento pela máquina



Linguagens de Representação do Conhecimento

- Uma *Linguagem de Representação do Conhecimento* é definida por:
  1) uma ***sintaxe***, que descreve as configurações que podem constituir sentenças daquela linguagem;
  2) uma ***semântica***, que liga cada sentença aos fatos no mundo que ela representa
     - cada sentença faz uma afirmação a respeito do mundo;
     - o agente acredita nas sentenças que correspondem a sua configuração interna.
- E tem um mecanismo de inferência associado = raciocínio

Representação & Raciocínio

- ***Raciocínio* é um processo de construção de novas sentenças a partir de sentenças existentes.**
- *Raciocínio "correto"* (***sound***):
  - Garante que as novas sentenças representam fatos, que se seguem de fatos representados pelas sentenças existentes na Base de Conhecimento (BC).
  - Implementa a relação de "*implicação*" entre sentenças

Linguagens de Representação do Conhecimento

- Linguagens de programação:
  - São precisas, porém não são suficientemente expressivas.
- Linguagens naturais:
  - São muito expressivas, porém são ambíguas.
- Linguagens de Representação de Conhecimento (LRC):
  - Utilizadas para expressar as sentenças das BC;
  - Existem 3 grandes classes:
    - Linguagens (predominantemente) declarativas;
    - Linguagens procedimentais;
    - Linguagens híbridas.

Meta-conhecimento

- Meta-conhecimento:
  - Conhecimento sobre o conhecimento disponível: escolha de ações
    - Atacar ou negociar?
  - *// Ente duas ações conflitantes, escolha a de maior utilidade*

Meta-Conhecimento

- É usado para aceder a conhecimento mais orientado para resolver determinado problema;
- Aumenta a eficiência de resolução do problema dirigindo o raciocínio para o subconjunto de conhecimento mais adequado;
- Representado através de **meta-regras** – regras que descrevem como usar outras regras.
- **Exemplo:**

  Se o carro não pega
  E o sistema elétrico está operacional
  Então usar regras relativas ao circuito de alimentação

Critérios de avaliação de LRC

- Expressividade
  - o que é possível dizer facilmente na linguagem?
- Inferência disponível:
  - que tipo de inferência é possível fazer na linguagem?
- Correção:
  - a inferência é plausível? A semântica é bem definida?
- Eficiência:
  - a inferência se realiza em um tempo razoável?
- Modularidade:
  - é fácil identificar e reutilizar partes do conhecimento?
- Legibilidade:
  - é fácil de ler e entender o que está escrito?
- Eficiência aquisicional:
  - é fácil adicionar conhecimento?

Solucionando o caso do cap. West (em LPO)

conhecimento prévio

A) $\forall x,y,z$ Americano(x) $\wedge$ Arma(y) $\wedge$ Nação(z) $\wedge$ Hostil(z) $\wedge$ Vende(x,z,y) $\Rightarrow$ Criminoso(x)
B) $\forall x$ Guerra(x,USA) $\Rightarrow$ Hostil(x)
C) $\forall x$ InimigoPolítico(x,USA) $\Rightarrow$ Hostil(x)
D) $\forall x$ Míssil(x) $\Rightarrow$ Arma(x)
E) $\forall x$ Bomba(x) $\Rightarrow$ Arma(x)
F) Nação(Cuba)
G) Nação(USA)
H) InimigoPolítico(Cuba,USA)
I) InimigoPolítico(Irã,USA)

conhecimento do problema

J) Americano(West)
K) $\exists x$ Possui(Cuba,x) $\wedge$ Míssil(x)
L) $\forall x$ Possui(Cuba,x) $\wedge$ Míssil(x) $\Rightarrow$ Vende(West, Cuba,x)

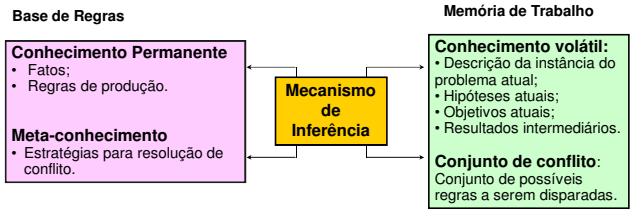
novo conhecimento

M) Possui(Cuba,M1)	*- Eliminação: quantificador existencial e conjunção de K*
N) Míssil(M1)	*- Modus Ponens a partir de D e N*
O) Arma(M1)	*- Modus Ponens a partir de C e H*
P) Hostil(Cuba)	*- Modus Ponens a partir de L, M e N*
Q) Vende(West,Cuba,M1)	*- Modus Ponens a partir de A, J, O, F, P e Q*
R) Criminoso(West)	

Representação de conhecimento: Regras de produção

- Representam conhecimento com pares de condição-ação
  - **SE** condição (ou premissa ou antecedente) ocorre **ENTÃO** ação (resultado, conclusão ou consequente) deverá ocorrer
    - **SE** o "semáforo" está verde **ENTÃO** a ação é seguir em frente
  - Em geral, uma regra pode ter múltiplos antecedentes ligados pelos conectivos lógicos E e OU (ou ambos);
  - O consequente de uma regra também pode ter múltiplas cláusulas.

Arquitetura dos Sistemas de Produção



Exemplo de regras para veículos

- Bicicleta: Se **veículoTipo=ciclo**
  E **num-rodas=2**
  E **motor=não**
  Então **veículo=*Bicicleta***
- Triciclo: Se **veículoTipo=ciclo**
  E **num-rodas=3**
  E **motor=não**
  Então **veículo=*Triciclo***
- Motocicleta: Se **veículoTipo=ciclo**
  E **num-rodas=2**
  E **motor=sim**
  Então **veículo=*Motocicleta***

Exemplo de regras para veículos

- CarroSport: Se **veículoTipo=automóvel**
  E **tamanho=pequeno**
  E **num-portas=2**
  Então **veículo=*CarroSport***
- Sedan: Se **veículoTipo=automóvel**
  E **tamanho=médio**
  E **num-portas=4**
  Então **veículo=*Sedan***
- MiniVan: Se **veículoTipo=automóvel**
  E **tamanho=médio**
  E **num-portas=3**
  Então **veículo=*MiniVan***

Exemplo de regras para veículos

- UtilitárioSport: Se **veículoTipo=automóvel**
  E **tamanho=grande**
  E **num-portas=4**
  Então **veículo=*UtilitárioSport***
- Ciclo: Se **num-rodas<4**
  Então **veículoTipo=*ciclo***
- Automóvel: Se **num-rodas=4**
  E **motor=sim**
  Então **veículoTipo=*automóvel***

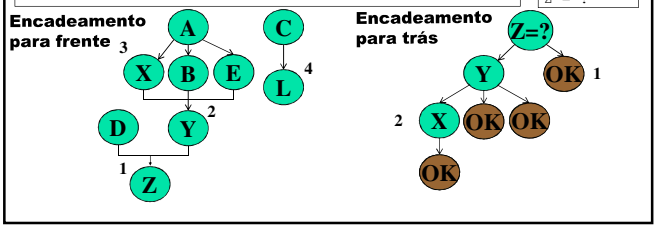
Complementando o exemplo...

- Meta-regras:
  - Se R1 e R2 podem ser disparadas, escolha R1;
  - Se R1 e R2 podem ser disparadas e R1 foi disparada mais recentemente que R2, escolha R2.
- Fatos:
  - Veículo1: tamanho=pequeno; num-portas=2; motor=sim
  - Veículo2: num-rodas=2; motor=não

Representação de conhecimento: um exemplo

```
Regra 01: Se Y = SIM & D = SIM Então Z = SIM
Regra 02: Se X = SIM & B = SIM & E = SIM Então Y = SIM
Regra 03: Se A = SIM Então X = SIM
Regra 04: Se C = SIM Então L = SIM
Regra 05: Se L = SIM & M = SIM Então N = SIM
```

```
A = SIM
B = SIM
C = SIM
D = SIM
E = SIM
Z = ?
```



Representação de conhecimento: regras de produção

- Raciocínio **progressivo** (encadeamento para a frente)
  - Dos dados à conclusão - ***data-driven inference***
  - As regras da BC são usadas para gerar informação nova (novos fatos) a partir de um conjunto inicial de dados
  - Os fatos gerados passam a fazer parte da BC
    - ex.: criminoso(West)
- Raciocínio **regressivo** (encadeamento para trás)
  - Da hipótese aos dados - ***goal-directed inference;***
  - Usa as regras da BC para responder a perguntas;
  - Prova se uma asserção é verdadeira
    - ex.: criminoso(West)?
  - Só processa as regras relevantes para a pergunta (asserção).
- Qual o melhor?

Representação de conhecimento: Regras de produção

- Resolução de conflitos:

```
Regra 01: Se Luz_Sinal = Verde    Então Ação = Continue
Regra 02: Se Luz_Sinal = Vermelho Então Ação = Pare
Regra 03: Se Luz_Sinal = Vermelho Então Ação = Continue
```

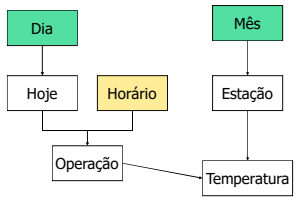
- O que fazer?
  - Parar quando o objetivo for alcançado;
  - Regra com maior prioridade;
  - Regra mais específica;
  - Regra mais recente;
  - Meta-conhecimento.

Representação: Regras de produção

- Vantagens:
  - As regras são de fácil compreensão;
  - Inferência e explicações são facilmente derivadas;
  - Manutenção é relativamente simples, devido a modularidade;
  - São mais eficientes que os sistemas de programação em lógica, embora menos expressivos.
- Desvantagens:
  - Conhecimento complexo requer muitas (milhares de) regras;
  - Esse excesso de regras cria problemas para utilização e manutenção do sistema;
  - Não são robustos (tratamento de incerteza);
  - Não aprendem.

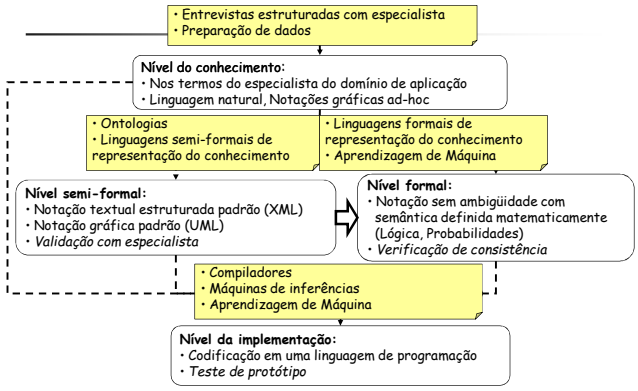
Sistemas Especialista: Termostato

- Objetivo: selecionar a temperatura adequada para o sistema de calefação
  - Mês do ano, dia da semana e horário do dia




```
Regra 01: Se Dia = Seg ou Dia = Ter ou Dia = Qua ou Dia = Qui ou Dia =
          Sex Então Hoje = DT
Regra 02: Se Dia = Sab ou Dia = Dom Então Hoje = FS
Regra 03: Se Hoje = DT & 9 < Horario < 17 Então Operação = DHT
Regra 04: Se Hoje = DT & Horario < 9 Então Operação = FHT
Regra 05: Se Hoje = DT & Horario > 17 Então Operação = FHT
Regra 06: Se Hoje = FS Então Operação = FHT
Regra 07: Se Mes = Jan ou Mes = Fev ou Mes = Dez Então estação = ver
Regra 08: Se Mes = Mar ou Mes = Abr ou Mes = Mai Então estação = out
Regra 09: Se Mes = Jun ou Mes = Jul ou Mes = Ago Então estação = Inv
Regra 10: Se Mes = Set ou Mes = Out ou Mes = Nov Então estação = Pri
Regra 11: Se estação = Pri e Operação = DHT Então Temp = 20
Regra 12: Se estação = Pri e Operação = FHT Então Temp = 15
Regra 13: Se estação = Ver e Operação = DHT Então Temp = 24
Regra 14: Se estação = Ver e Operação = FHT Então Temp = 27
Regra 15: Se estação = Out e Operação = DHT Então Temp = 20
Regra 16: Se estação = Out e Operação = FHT Então Temp = 16
Regra 17: Se estação = Inv e Operação = DHT Então Temp = 18
Regra 18: Se estação = Inv e Operação = FHT Então Temp = 15
```

Engenharia de uma base de conhecimento



Raciocinando com Encadeamento progressivo

(forward chaining inference engine)

Dos dados à conclusão

data-driven inference

Raciocinando com Encadeamento Progressivo

- Dos dados à conclusão
  - Parte dos fatos na BR (Base de Regras) e na memória de trabalho, buscando quais regras eles satisfazem, para produzir assim novas conclusões (fatos) e/ou realizar ações.
- Três etapas:
  - Busca, Casamento (unificação), Resolução de conflito
- É uma estratégia de inferência muito rápida
  - Utilizada em sistemas de monitoramento e diagnóstico em tempo real.
- Ferramentas comerciais que implementam esta estratégia
  - OPS5, OPS83, IBM: TIRS

Encadeamento progressivo Algoritmo

1. Armazena as regras da BR (base de regras) na máquina de inferência (MI) e os fatos na memória de trabalho (MT);
2. Adiciona os dados iniciais à memória de trabalho;
   obs.: Esses dados devem ser fornecidos pelo usuário do sistema
3. Compara o antecedente das regras com os fatos na MT
   - Todas as regras cujo antecedente "casa" (unifica) com esses fatos podem ser disparadas, e são colocadas no conjunto de conflito.
4. Usa o procedimento de resolução de conflito para selecionar uma única regra desse conjunto;
   - Se o conjunto de conflito estiver vazio, o algoritmo para;
   - A MT é apresentada ao usuário.
5. Dispara a regra selecionada, atualiza a MT & volta ao passo 3.

Encadeamento progressivo Busca e Casamento (unificação)

- O algoritmo tenta casar (unificar) as premissas das regras selecionadas com os fatos na memória de trabalho
  - MT1: num-rodas=4, motor=sim, num-portas=3, tamanho=médio
  - MI (regras da BC): Se num-rodas=4 E motor=sim
    Então veículoTipo=automóvel
  - MT2: MT1 + veículoTipo= automóvel

Encadeamento progressivo: Resolução de conflitos

- Resolução de conflitos:
  - Heurística geral para escolher um subconjunto de regras a disparar
- Exemplos:
  - Não duplicação: não executar a mesma regra com os mesmos argumentos duas vezes;
  - Prioridade de operação: preferir ações com prioridade maior;
  - Recency ("recenticidade"): preferir regras que se referem a elementos da Memória de Trabalho criados recentemente;
  - Especificidade: preferir regras que são mais específicas.

Encadeamento progressivo: Exemplo no domínio dos veículos

- Carregar a BR de veículos na 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 primeira 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 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*

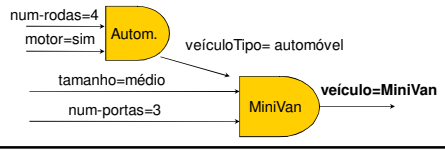
Encadeamento progressivo: Exemplo no domínio dos veículos

- 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 irá parar.
- 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*



Raciocinando com Encadeamento Regressivo

(backward chaining inference engine)

Da hipótese aos dados

goal-directed inference

Encadeamento Regressivo: Busca e Casamento

- Da hipótese aos dados :
  - Parte da hipótese que se quer provar, procurando regras na BR cujo consequente satisfaz essa hipótese;
  - Utiliza as regras da BR (base de regras) para responder às perguntas;
  - Busca provar se uma pergunta é verdadeira
    - ex.: West = criminoso?
  - 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 consequente 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)

Encadeamento Regressivo: Algoritmo

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)

   c) Quando todos os antecedentes são provados:
   - Dispara a regra = instancia a variável no seu consequente para o valor que aparece nessa regra e
   - Devolve o resultado ao usuário (o algoritmo termina com sucesso).

Encadeamento Regressivo: Algoritmo

6. continuação:

   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 atualiza a pilha de objetivos, com base na nova variável objetivo – recursão!)

   - Se conseguir provar que o valor-verdade dessa nova variável é V:
     - Dispara a regra, instancia a variável no seu consequente para o valor que aparece nessa regra;
     - Retoma o processamento da regra que estava sendo provada antes (6.a)

Encadeamento Regressivo: Algoritmo

6d. Continuação:
- Se o valor-verdade dessa nova variável é F:
  - Abandona a regra e volta para a pilha de objetivos;
  - Se a pilha de objetivos estiver vazia, o algoritmo falha.

- 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 tendo essa variável como variável objetivo.
    (recursão de novo!)

Encadeamento Regressivo: Busca e Casamento

- O sistema percorre a BC em busca regras cujo consequente "casa" com a hipótese de entrada
  - unificação é realizada com busca em profundidade
- Não há conjunto de conflito de regras mas uma pilha de objetivos

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 a variável objetivo no consequente
    - 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 consequente;
    - 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 não é trivial;
  - Só uma abordagem probabilista pode tratar este tipo de incerteza corretamente (e.g. redes bayesianas)

The End