

DCA0121 – INTELIGÊNCIA ARTIFICIAL APLICADA
Aula 6 – Representação do Conhecimento e Sistemas
Especialistas

Prof. Marcelo Augusto Costa Fernandes
mfernandes@dca.ufrn.br

Introdução

- RC – Representação do Conhecimento
- KR - *Knowledge Representation*
- Teoria e prática da representação do conhecimento em sistemas computacionais

Características

- Em IA RC é focado na resolução de problemas com as seguintes características:
 - O conhecimento deve ser armazenado
 - Devem existir formas para buscar e associar conhecimentos
 - Finalmente, deve-se gerar novos conhecimentos a partir de outros
- A principal regra é habilitar uma entidade inteligente dotada de uma base de conhecimento, tomar decisões inteligentes em um ambiente.

Conhecimento e RC

- O conhecimento pode ser associado a o fato em si, ou em outras palavras, verdades em algum mundo relevante;
- Representação do Conhecimento
 - O fato é representado por algum formalismo conhecido para poder ser manipulado por um sistema computacional.
 - Alguns tipos de formalismo:
 - Objetos,
 - Proposições e definições,
 - Conceitos e relações,
 - Teoremas e regras,
 - Algoritmos,
 - Estratégias e táticas,
 - Metaconhecimento (conhecimento sobre o conhecimento),
 - ...

Representação do Conhecimento

- Algumas definições
 - Formalização e estruturação de um conhecimento real em um sistema computacional, para gerar um comportamento inteligente.
 - A RC deve respeitar alguma semântica formal
 - A semântica é a ponte entre o mundo real e a representação formal do conhecimento

Alguns termos

- Base de conhecimento
 - Conjunto de regras, heurísticas, fatos e dados que compõe um dado conhecimento
 - Pode ou não ser atualizada
 - Deve permitir geração de novos fatos ou regras com base nas existentes
- Inferência
 - A inferência é a etapa na qual novos conhecimentos são deduzidos tomando como referência a base de conhecimento.
- Máquina de inferência
 - Executa os algoritmos e regras que processam a inferência

Abordagens para RC

- Declarativa ou lógica
 - O conhecimento é formalizado com fatos postulados como verdadeiros
 - O conhecimento é formalizado com proposições declarativas
 - Existem um conjunto regras para manipular o conhecimento
 - Exemplo:
 - A representação do conhecimento pode ser feita utilizando a lógica
- Procedimental
 - O conhecimento pode ser formalizado por procedimentos ou regras de como desempenhar tarefas
 - Facilidade na representação computacional
 - Exemplo:
 - Regras de produção

Abordagens para RC

- Estrutural
 - A descrição do conhecimento é feita com objetos e relações entre as entidades
 - São utilizados conceitos de hierarquias, classes e instâncias
 - A base de conhecimento é associada a coleção de objetos e suas relações
 - O aspecto estrutural é importante na caracterização do conhecimento.
 - Exemplo: Frames, POO

Métodos de Representação do Conhecimento

- Um método de representação do conhecimento pode ser definido com um esquema de formalização e estruturação do conhecimento
- A escolha do método depende muito tipo do problema e o tipo de solução
- Existem vários métodos e técnicas de representação do conhecimento

Método RC baseado em Lógica

- Utiliza conhecimento declarativo, expresso por qualquer tipo de lógica:
 - Lógica proposicional (Lógica sentencial)
 - Lógica de primeira ordem (LPO)
 - Lógica temporal
 - Alguns fatos são verdadeiro em momentos particulares
 - Lógica multi-valorada
 - Lógica nebulosa (fuzzy)

Lógica Proposicional (*PL - Propositional Logic*)

- Formas mais primitivas de representação do conhecimento
- Forma mais comum da lógica
- Tem como base proposições que podem ser verdadeiras ou falsas
- As proposições são ligadas pelos conectores ou operadores lógicos

Lógica Proposicional

- Conectores lógicos
 - \wedge (e);
 - \vee (ou);
 - \rightarrow (se então);
 - \leftrightarrow (se e somente se);
 - \neg (não);

Lógica Proposicional

- Vantagem na representação formal propiciada pela lógica é a derivação de novos fatos a partir de fatos já conhecidos como certos
- Regra de inferência
 - Função sintática que, dado um conjunto de fórmulas lógicas, gera uma nova fórmula. “
 - Raciocínio baseado no conhecimento que já possui, gerando novos.
 - Exemplo
 - *Silogismo Hipotético*
 - $((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$

Lógica de Primeira Ordem ou Lógica de Predicados

- *FOP – First Order Logic* ou *PL – Predicate Logic*
- Extensão da lógica proposicional
- Introduz os operadores de quantificação
 - \forall (quantificador universal - para todo);
 - \exists (quantificador existencial - existe um)
- Na PL a validade das proposições não considera os predicados (conceito, atributo ou relação) que um determinado objeto possui
- Trabalha com os objetos e seus predicados
- Mais utilizada na representação de conhecimento declarativo

Lógica de Primeira Ordem ou Lógica de Predicados

- Fatos
 - Predicados e argumentos;
- Predicados
 - Relacionamento entre os objetos ou o nome dos atributos dos objetos
- objetos
 - Argumentos dos predicados;
- Exemplo
 - “todos os homens são mortais” e “Sócrates é homem” podem ser representadas como:
 - $\forall x (\text{Homem}(x) \rightarrow \text{Mortal}(x))$
 - $\text{Homem}(\text{Sócrates})$
 - \forall : quantificador universal;
 - Homem e Mortal são predicados;
 - x é uma variável;
 - Sócrates é uma constante;

Lógica de Primeira Ordem ou Lógica de Predicados

- Algumas vantagens
 - Representação formal que possui um conjunto de regras de inferência através do qual, a partir de fatos iniciais se pode derivar outros fatos
 - Pode-se garantir a verdade dos novos fatos deduzidos formalmente, se os fatos iniciais forem verdadeiros
- Algumas desvantagens
 - Inadequado para inferir novos dados a partir de dados incertos, valores relativos tipo “muito quente”.

Lógica de Primeira Ordem ou Lógica de Predicados

- Exemplo de representações:
 - Marcos era um homem
 - Homem (Marcos)
 - Marcos nasceu em Pompéia
 - Pompeano (Marcos)
 - Todos os que nasceram em Pompéia eram romanos
 - $\forall x: \text{Pompeano}(x) \rightarrow \text{Romano}(x)$
 - César era um soberano
 - soberano (César)
 - Todos os romanos eram leais a César ou então odiavam-no
 - $\forall x: \text{Romano}(x) \rightarrow \text{leal_a}(x, \text{César}) \vee \text{odeia}(x, \text{César})$

Regras de Produção (Production Rules ou Production Systems)

- Representação baseada em regras
 - O conhecimento é representado por regras que são formadas por duas partes:
 - uma condição (se) e uma ação (então)
- O conhecimento é representado como uma coleção de regras do tipo *se condição então ação*
 - A ação corresponde a algum procedimento que acarreta uma conclusão ou mudança no estado corrente;
- Exemplo de sistema de representação procedimental

Regras de Produção

- Propriedades
 - Incorporam conhecimento prático (heurístico) em regras se-então
 - Sua habilidade cresce em uma taxa proporcional ao crescimento da base de conhecimento (crescimento incremental)
 - Pode resolver um grande intervalo de problemas possivelmente complexos utilizando regras relativamente simples e combinando os resultados de maneira apropriada;
 - Podem explicar suas conclusões refazendo suas linhas de raciocínio e traduzindo a lógica de cada regra empregada em linguagem natural

Regras de Produção

- Um sistema típico de regras de produção é formado por:
 - uma base de conhecimento;
 - memória de trabalho;
 - máquina de inferências;
- Linguagens como Prolog e Lisp já incorporam a memória de trabalho e a máquina de inferências:
 - o que não ocorre em linguagens tradicionais como C e Java onde há necessidade de implementar estes mecanismos;
- Existem shells (programas específicos para implementação de regras) que já incorporam a memória de trabalho, a máquina de inferências e as rotinas de interface com o usuário

Regras de Produção

- Base de conhecimento:
 - composta por regras e fatos;
 - regras: declaração sobre classes e objetos:
 - SE condição (antecedente) ENTÃO ação (conseqüente);
 - fatos: declarações sobre objetos específicos;
- Memória de trabalho:
 - Representa o estado do problema em um dado instante (permite comunicação entre regras);
 - Possui dados dinâmicos de curta duração que existem enquanto uma regra estiver sendo interpretada

Regras de Produção

- Máquina de Inferência
 - É ativada quando existe um estado “meta”
 - Responsável pela execução das regras, determinando quais são relevantes, de acordo com a configuração da memória de trabalho e pela escolha de quais aplicar
 - O processo de inferência pode ser dividido em três fases:
 - Seleção de regras;
 - Objetivo é encontrar todas as regras que são satisfeitas pelo conteúdo da memória de trabalho gerando um conjunto conflito
 - Regras que satisfazem as “condições de casamento”, ou seja, regras passíveis de serem executadas

Regras de Produção

- Mecanismos de Inferência
 - As estratégias mais utilizadas para este casamento são:
 - Raciocínio para frente ou Encadeamento Progressivo (*Forward chaining*)
 - Raciocínio para trás ou encadeamento regressivo (*Backward chaining*)

Mecanismos de Inferência – Regras de Produção

- Encadeamento Progressivo (*Forward chaining*)
 - Raciocínio dirigido por regras
 - Estratégia que inicia com fatos ou informações conhecidas nos quais são derivados novos fatos. Este processo é repetido até que a solução seja encontrada ou nenhuma outra regra possa ser disparada.
 - Vantagens:
 - Deriva uma grande quantidade de informações a partir de poucos fatos
 - Desvantagens:
 - Pode-se fazer muitas perguntas
 - Pode-se derivar informações desnecessárias tornando o processo de inferência mais lento

Exemplo: Encadeamento Progressivo

Regra 01: **Se** B & C **Então** F

Regra 02: **Se** C & D **Então** G

Regra 03: **Se** A & F **Então** H

Regra 04: **Se** A & G **Então** I

Regra 05: **Se** E & G **Então** H

A = V

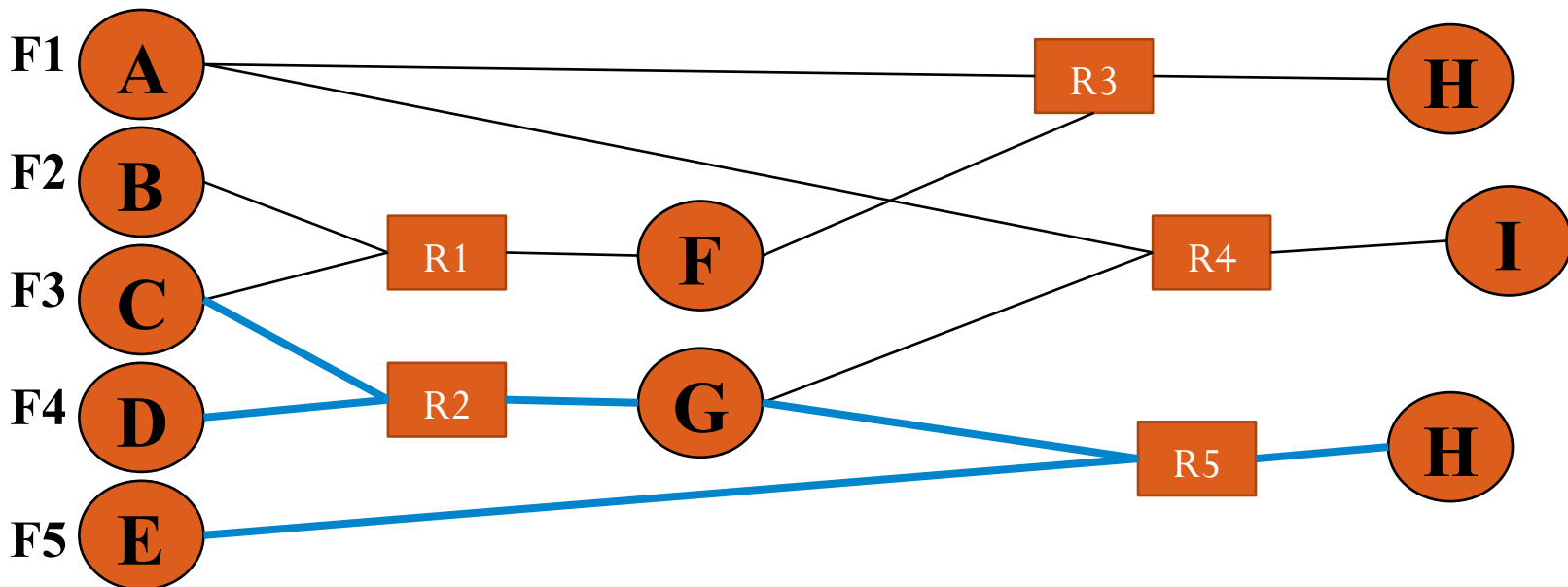
B = V

C = V

D = V

E = V

H = ?



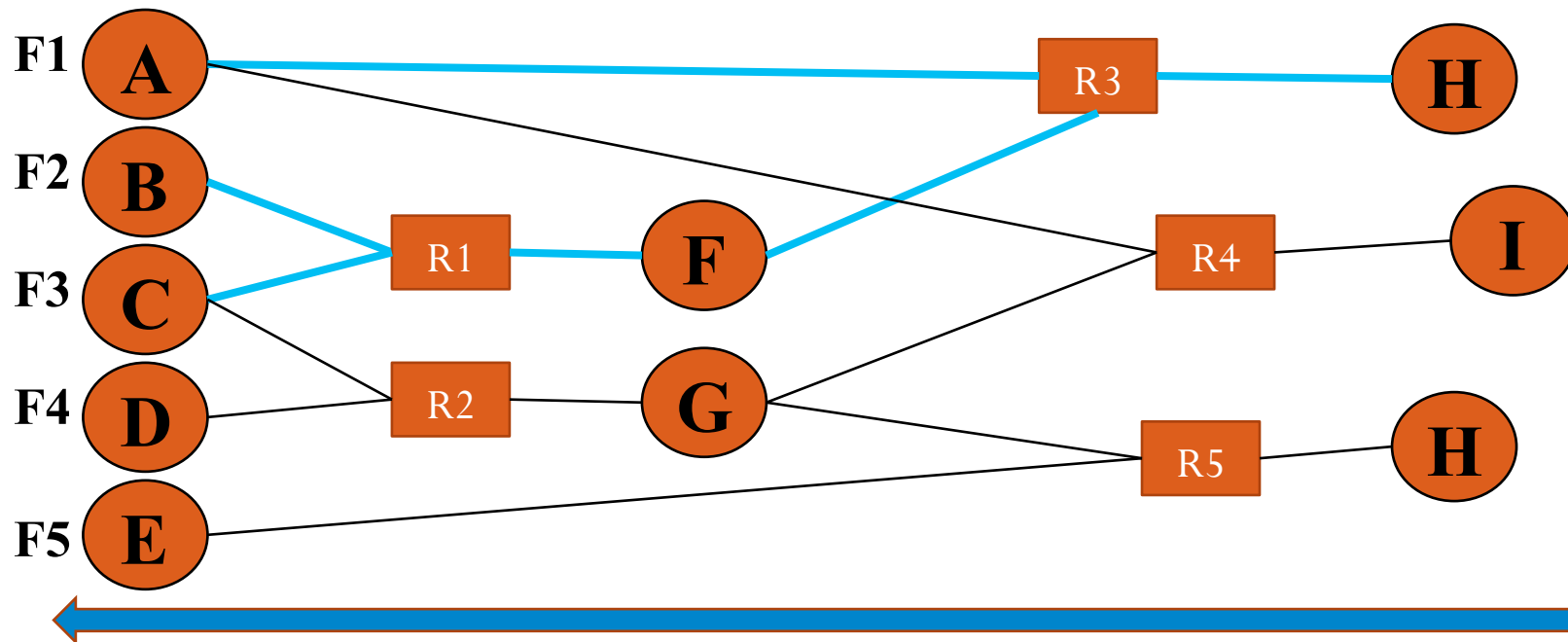
Mecanismos de Inferência – Regras de Produção

- Encadeamento regressivo (*Backward chaining*)
 - Raciocínio orientado por metas
 - Deve-se inicialmente definir o objetivo e procura as hipótese que definem este objetivo
 - Vantagens:
 - Deriva-se apenas informações relevantes ao problema
 - Desvantagens:
 - O sistema pode ficar lento quando o número de hipóteses é alto.

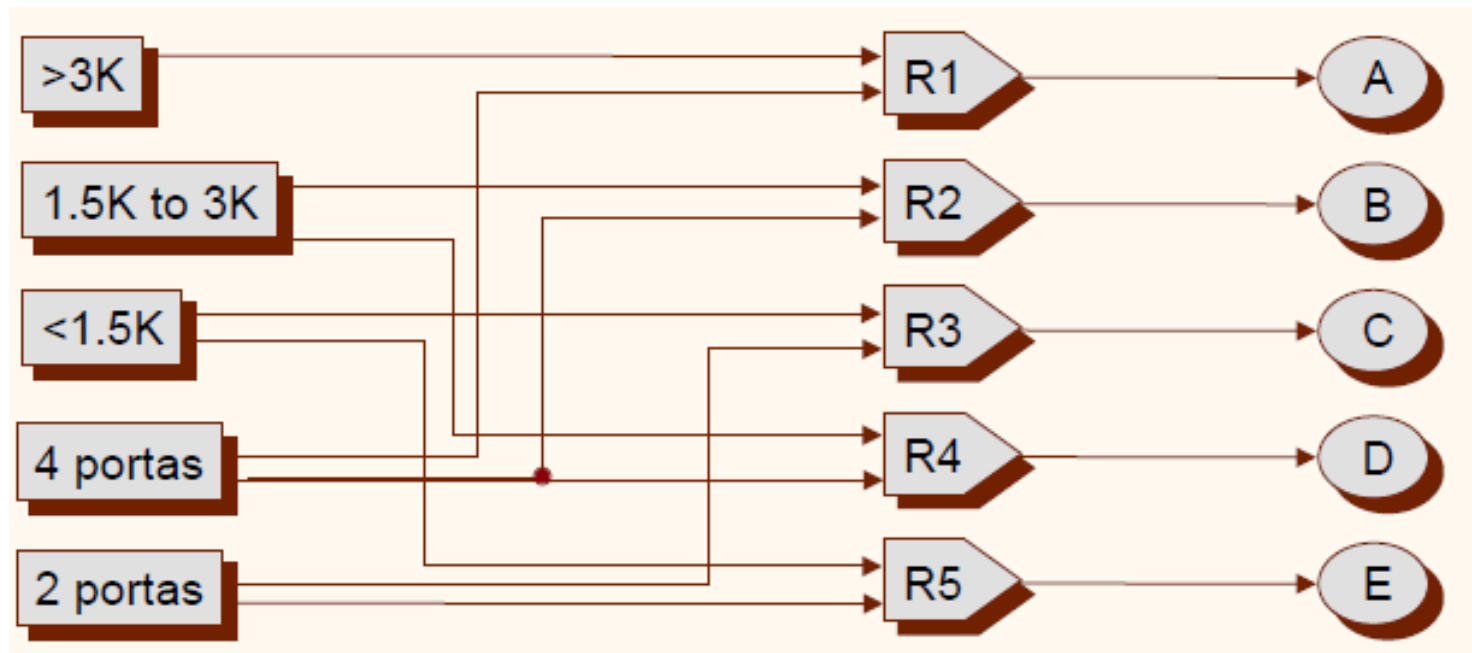
Exemplo: Encadeamento regressivo

Regra 01: **Se** B & C **Então** F
Regra 02: **Se** C & D **Então** G
Regra 03: **Se** A & F **Então** H
Regra 04: **Se** A & G **Então** I
Regra 05: **Se** E & G **Então** H

A = V
B = V
C = V
D = V
E = V
H = ?



Exemplo: Seleção de um automóvel

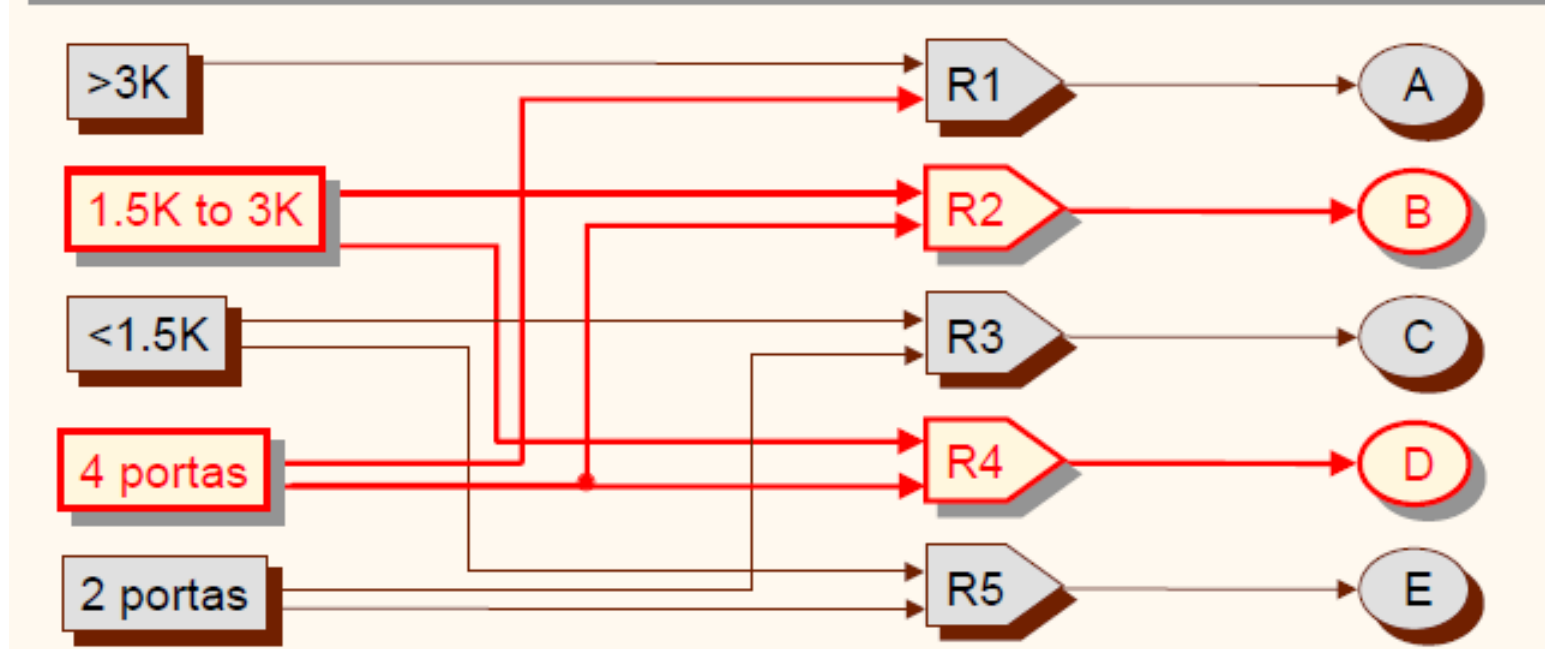


Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

Exemplo: Encadeamento Progressivo

Utilizador: Quero um automóvel de 4 portas que custe entre 1,5K e 3K; que automóvel devo comprar?

SP: Escolher entre automóveis B e D

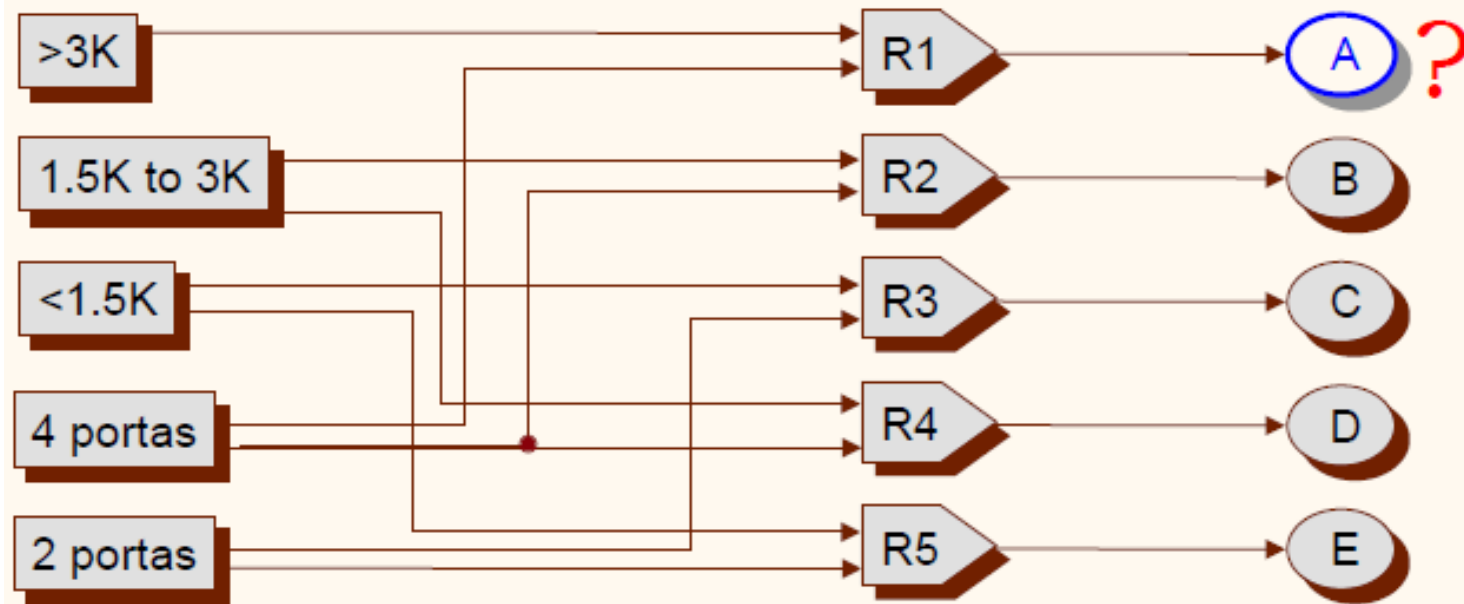


Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

Encadeamento regressivo

Utilizador: Quero um automóvel; qual devo comprar?

SP: ("pensando") Vamos tentar o automóvel A...

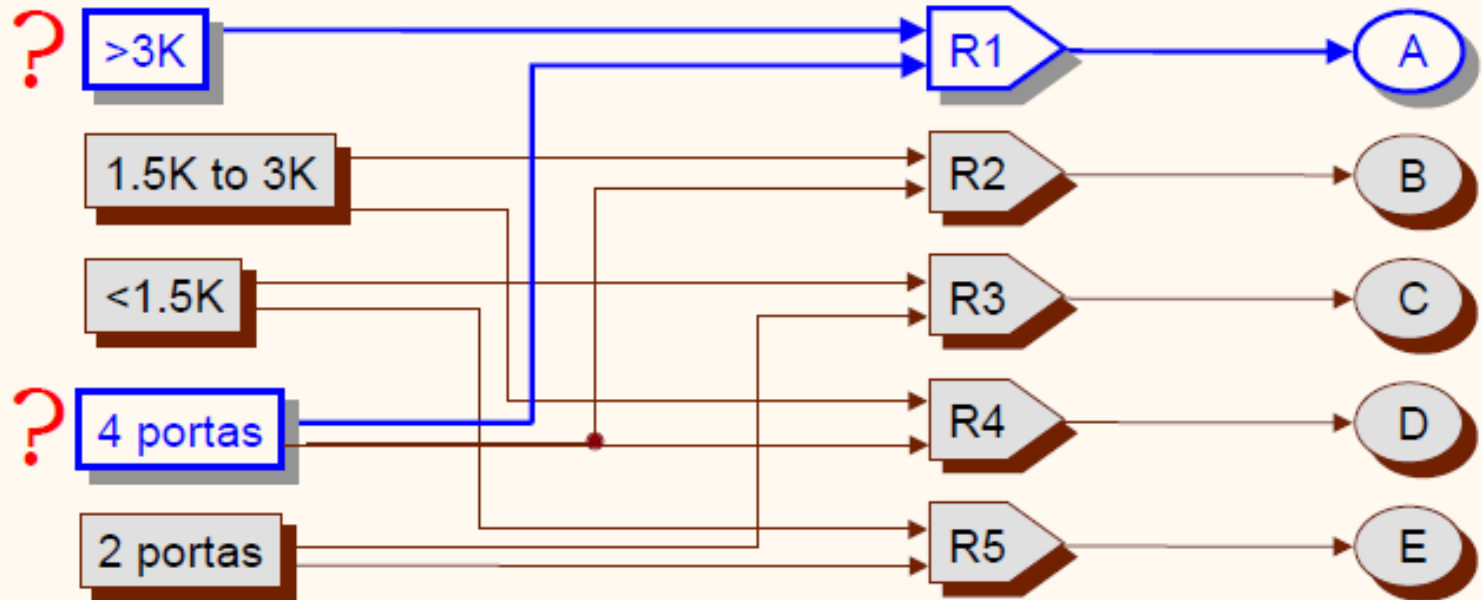


Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

Encadeamento regressivo

SP: *Pode gastar mais de 3K?*

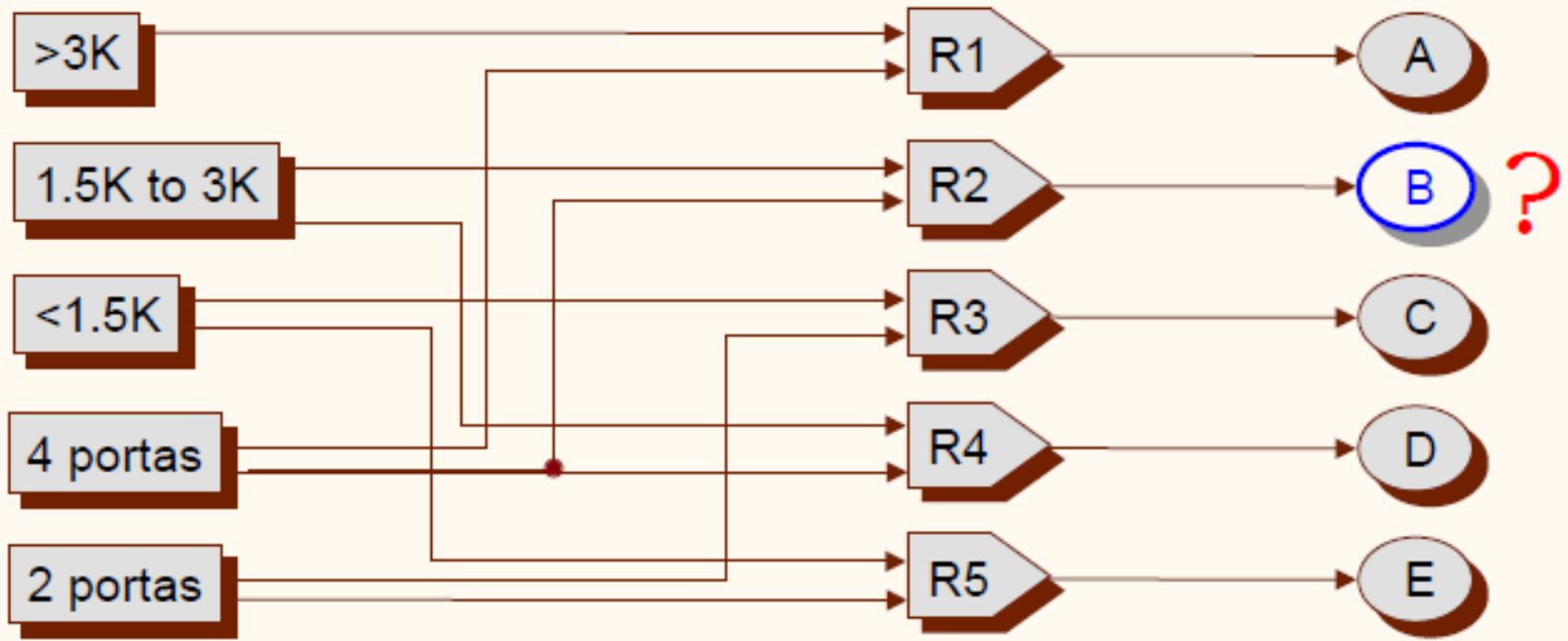
Utilizador: *Não*



Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

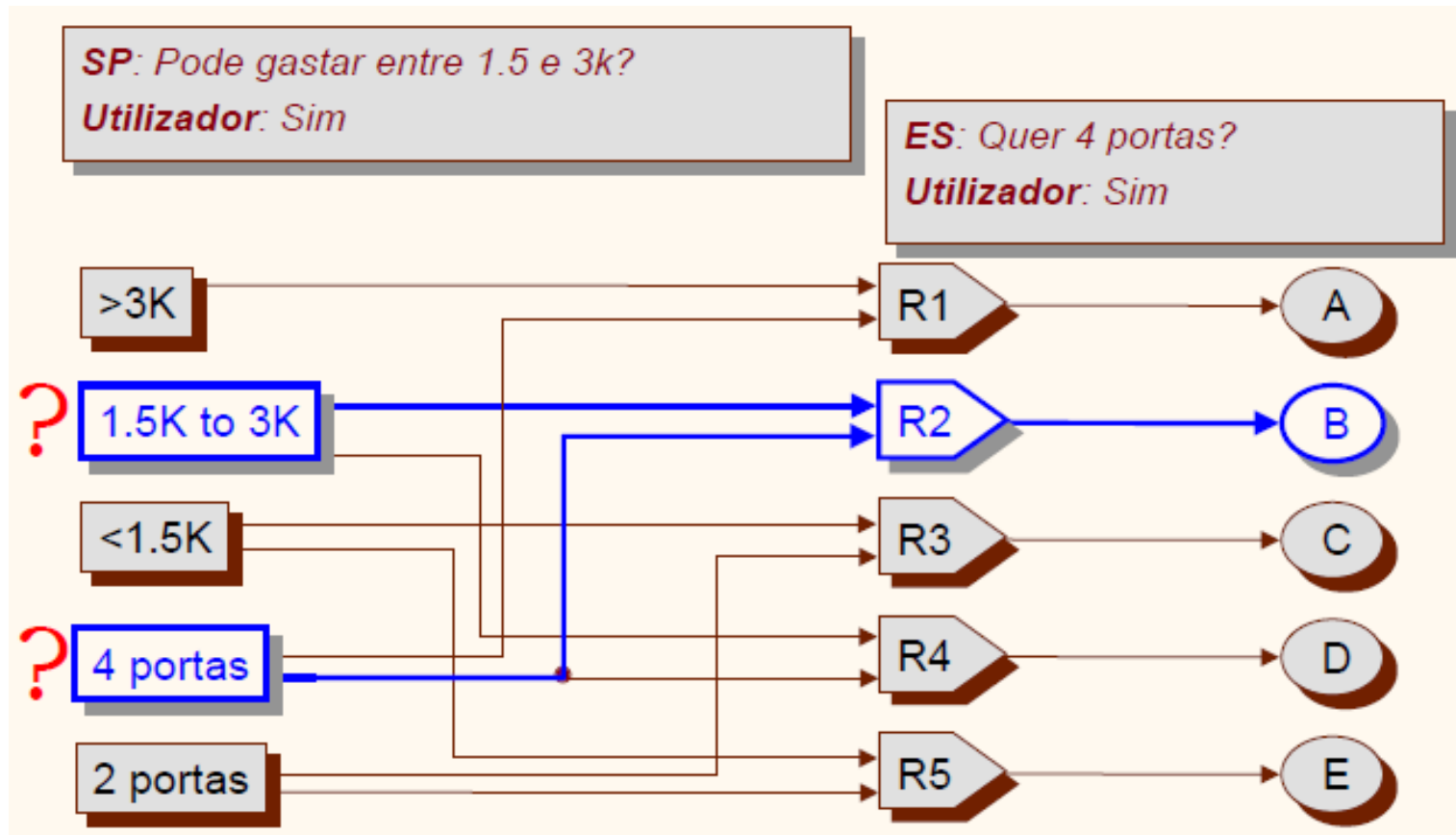
Encadeamento regressivo

SP: ("pensando") Hum... Vamos tentar o automóvel B



Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

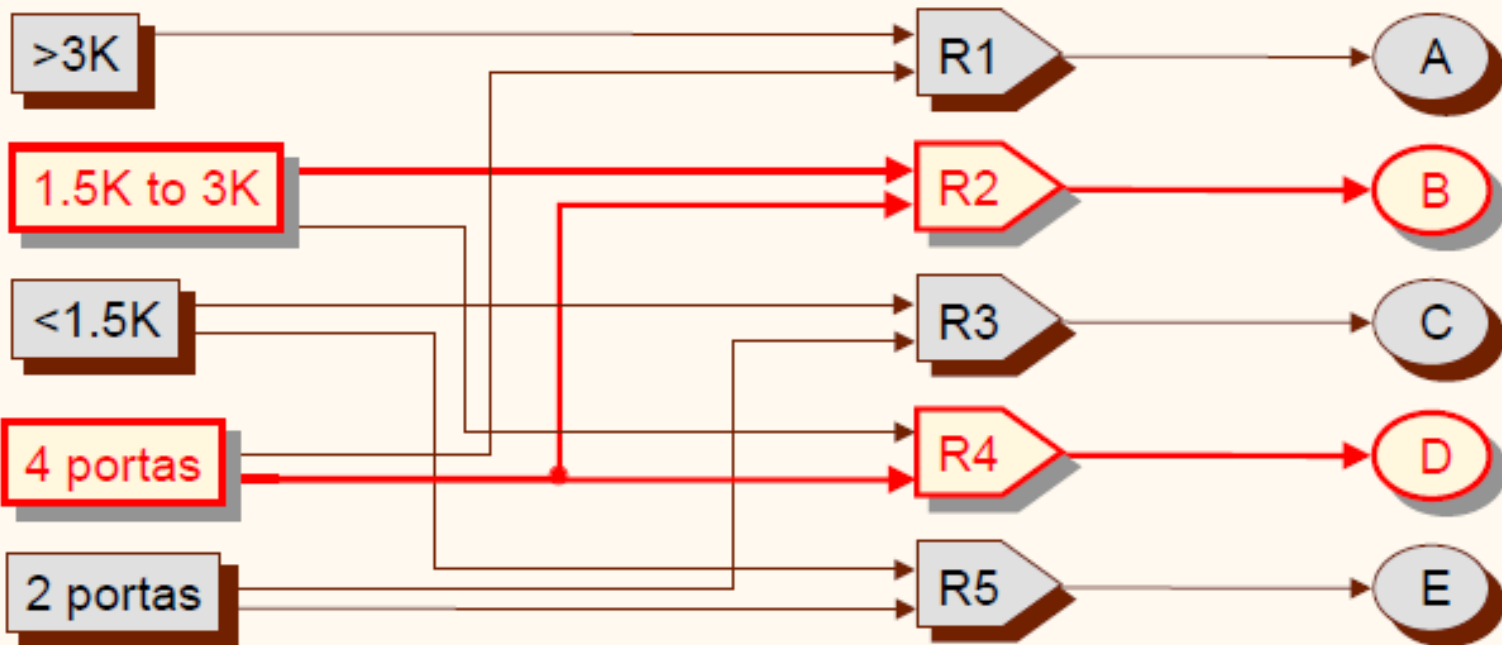
Encadeamento regressivo



Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

Encadeamento regressivo

SP: Escolher entre automóveis B e D



Fonte: <http://eden.dei.uc.pt/~ec/teoricas/ConInf29-80.pdf> (Acessado em 10/10/2012)

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
```

- Algumas técnicas:

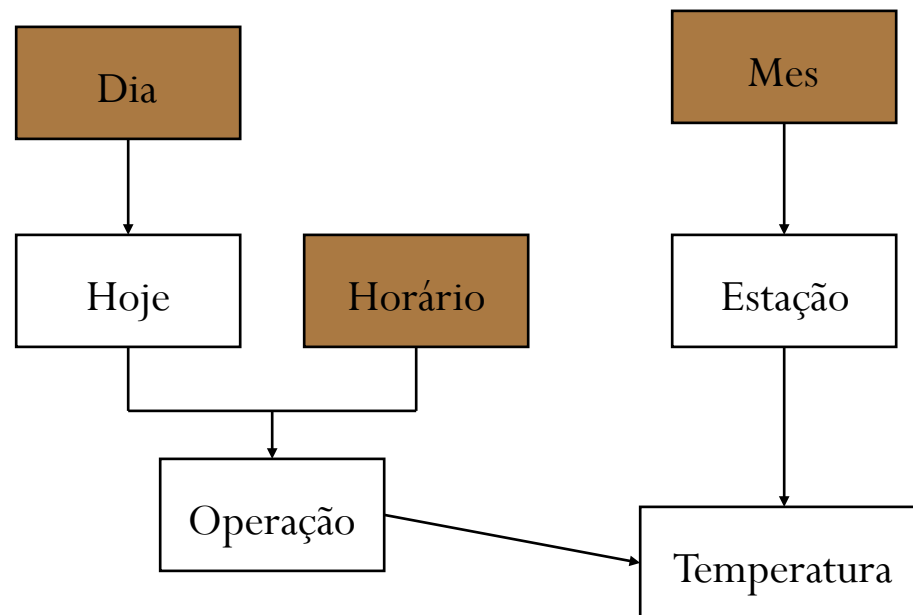
- Parar quando o objetivo for alcançado
- Regra com maior prioridade
- Regra mais específica
- Regra mais recente
- Meta-conhecimento

Regras de Produção

- Vantagens
 - As regras são de fácil compreensão.
 - Inferência e explicações são facilmente derivadas.
 - Manutenção é relativamente simples, devido a modularidade.
- 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

Regras de Produção

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



Regras de Produção

- 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 < Horário < 17 **Então** Operação = DHT
- Regra 04: **Se** Hoje = DT & Horário < 9 **Então** Operação = FHT
- Regra 05: **Se** Hoje = DT & Horário > 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

Técnicas de KR - Redes Semânticas (*SN - Semantic Networks*)

- Representação de conhecimentos de vários objetos, através da relação entre os mesmos
- Composta de vários nós (objetos) associados por arcos que representam a associação entre os mesmos
 - As ligações entre os objetos representam fatos
- Novos fatos podem ser inseridos ou retirados da SN proporcionando o desenvolvimento do conhecimento
- As relações entre os nós podem ser
 - *é_um* (*is_a*)
 - *tipo_de* (*AKO - A Kind Of*)

Técnicas de KR - Redes Semânticas (SN - Semantic Networks)

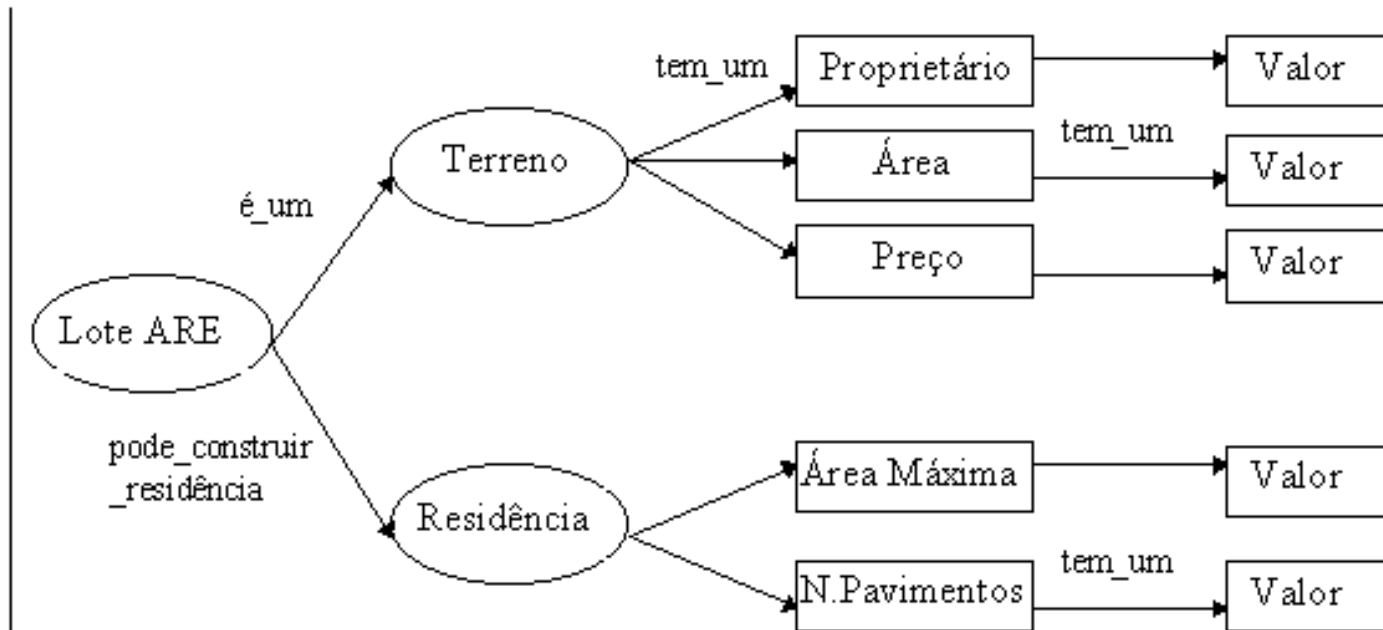
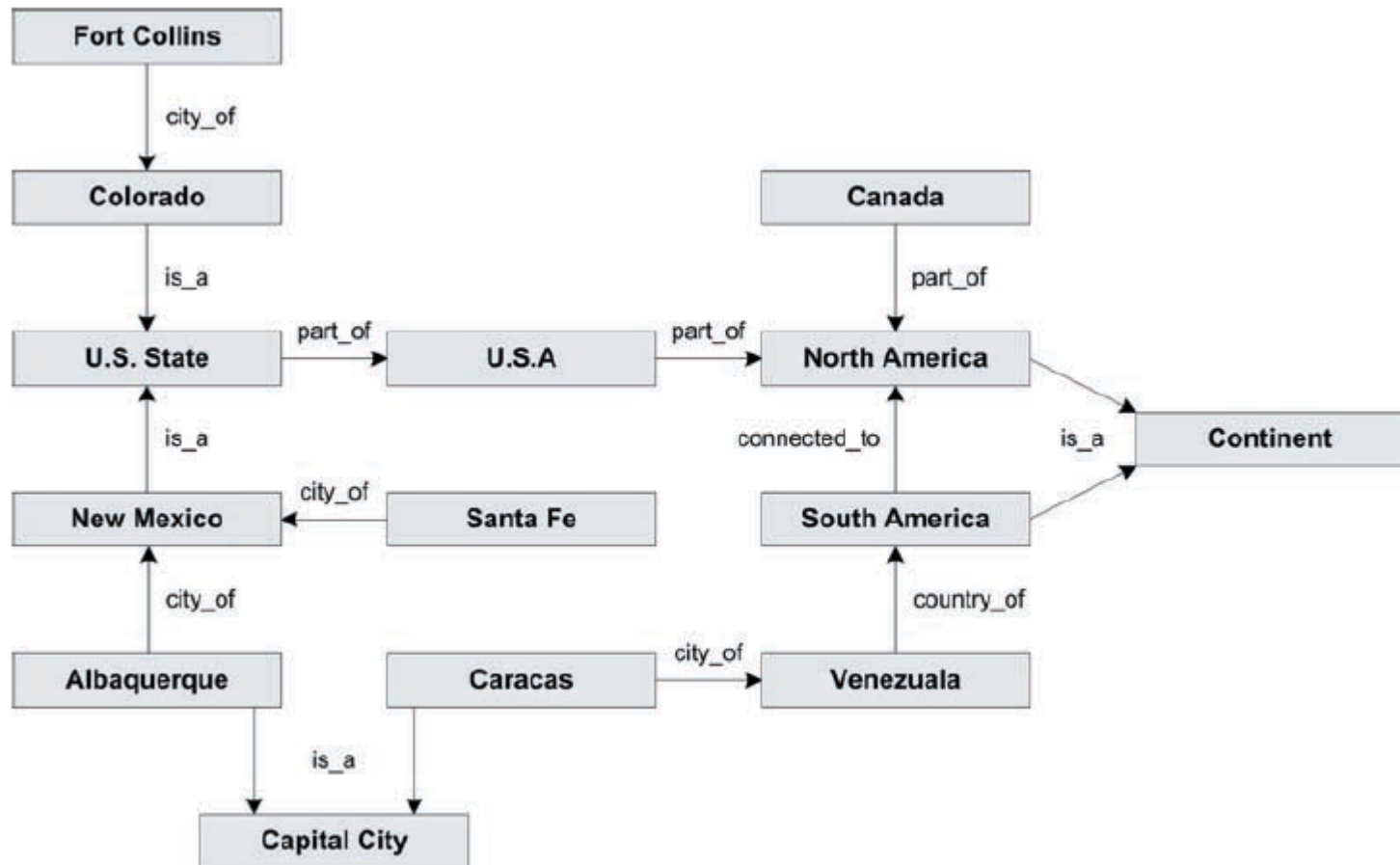


Fig. 2.5 Representação de Rede Semântica

Técnicas de KR - Redes Semânticas (*SN - Semantic Networks*)



Técnicas de KR - Redes Semânticas (SN - *Semantic Networks*)

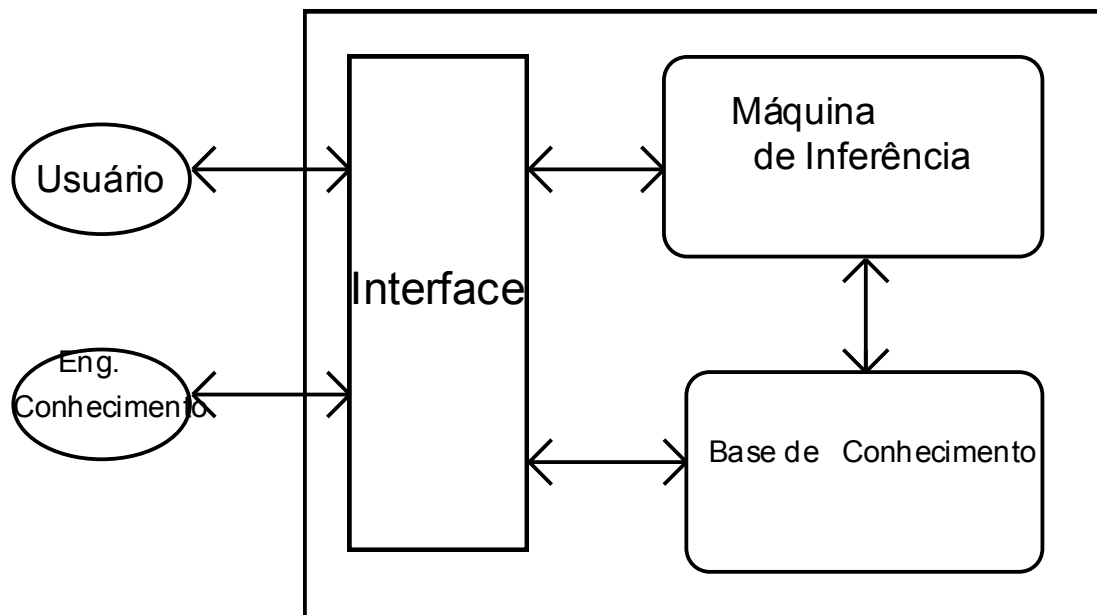
- Algumas vantagens
 - Grande flexibilidade na base associada a representação do conhecimento
 - Simplicidade na representação
 - Permite herança
 - Bastante associada a POO
 - Diagrama de Objetos da UML
- Algumas desvantagens
 - A busca a um determinado problema pode levar a visitação de muitos nós, dependendo do domínio do problema

Técnicas de KR - Redes Semânticas (*SN - Semantic Networks*)

- Algumas derivações
 - Web semântica

Sistema Baseado em Conhecimento - SBC

- É um sistema computacional que expressa o conhecimento sobre um determinado domínio de maneira explícita



Sistema Baseado em Conhecimento - SBC

- **Máquina de Inferência**

- É um processo fixo, algorítmico, baseado num processo de dedução lógica responsável pela extração de todos os resultados possíveis de serem obtidos a partir do conhecimento armazenado na BC.

- **Base de Conhecimento - BC**

- É a parte que armazena o conhecimento de uma determinada área do conhecimento.
- Toda a *programação* de um SBC é feita através do armazenamento do conhecimento sobre o problema, expresso por meio de formalismos de representação de conhecimento, na BC.
- As formas de representação do conhecimento mais utilizadas são:
 - regras de produção (se...então)
 - Frames
 - ...

Sistemas Especialistas

- Os sistemas especialistas formam um subconjunto dos SBCs por armazenarem na BC o conhecimento empírico de um profissional especializado em algum domínio.
- Este conhecimento difere dos demais pela simples razão de não estar normalmente escrito ou expresso em manuais, livros, etc e por chegar a solução de maneira mais eficiente.
- Este conhecimento tem por origem a experiência que o profissional adquire durante a resolução de problemas em um determinado domínio.
- Forma um conjunto de métodos que fazem dele um especialista.

Sistemas Especialistas

Sistemas Desenvolvidos com técnicas de IA

Sistemas Baseados em Conhecimento

Sistemas Especialistas

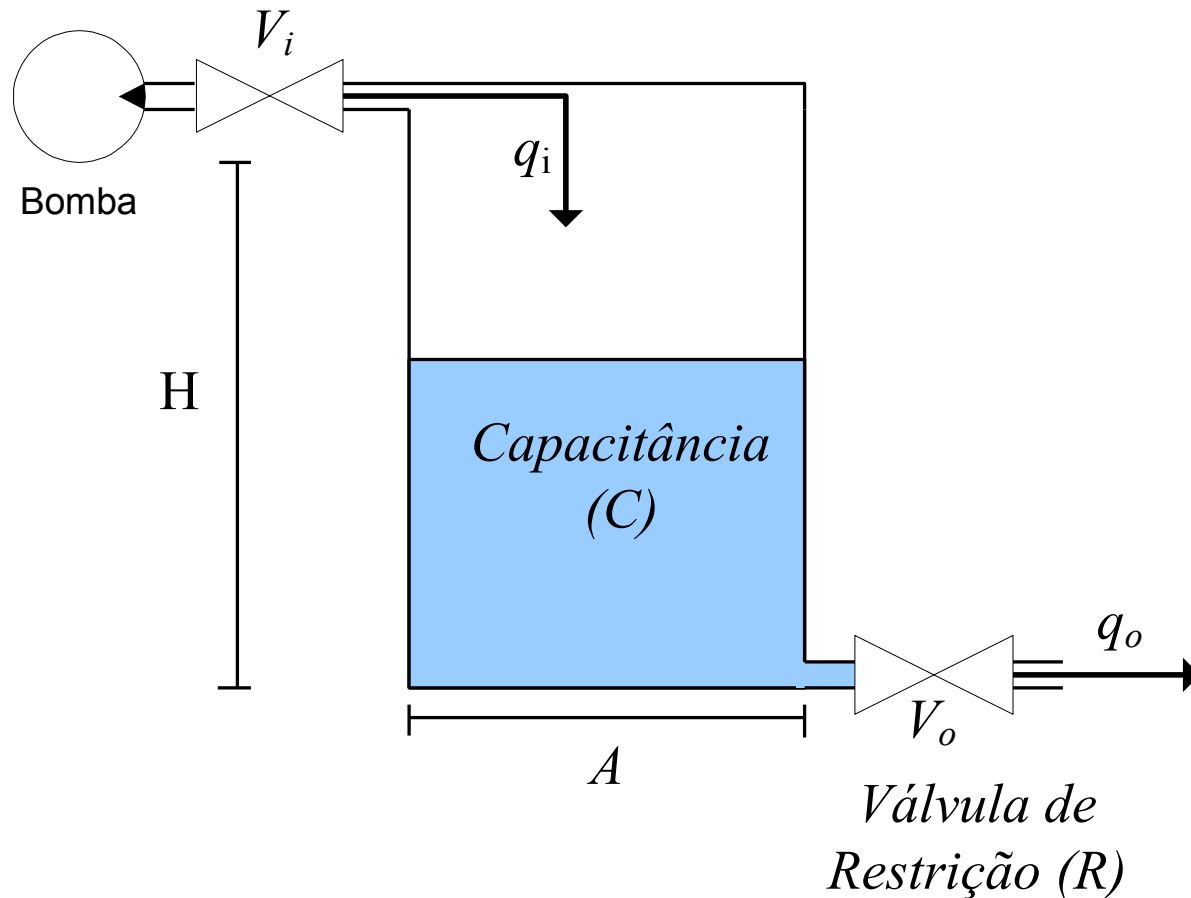


Tópicos sobre Prolog

- É uma linguagem orientada ao processamento simbólico;
- Representa uma implementação da lógica como linguagem de programação;
- Apresenta uma semântica declarativa inerente à lógica;
- Permite a obtenção de respostas alternativas;
- Suporta código recursivo e iterativo para a descrição de processos e problemas, dispensando os mecanismos tradicionais de controle, tais como *while*, *repeat*, *etc*;
- Permite associar o processo de especificação ao processo de codificação de programas;
- Representa programas e dados através do mesmo formalismo;
- Incorpora facilidades computacionais de metalógicas.
- Trabalha na linguagem natural.

Exemplos: Projeto 1

- Controle de Nível de um Tanque



Projeto 1: Requisitos

- As válvulas V0 e V1 são digitalmente controladas onde um pulso 1 abre totalmente a válvula e um pulso 0 fecha totalmente a válvula.
- O tanque possui quatro sensores digitais S0, S1 e S2. Estão sensores transmitem 1 na presença da água e 0 quando não á presença de água.
- A Válvula V0 enche o tanque e a válvula V1 seca o tanque.
- O sensor S2 esta no topo do tanque, o S1 no meio e S0 no fundo do tanque.
- O sistema deve possui uma chave on/off para iniciar o processo de controle.
 - O objetivo é manter o nível de água entre os sensores S2 e S1 após a chave for ligada.
 - Quando a chave for desligada o tanque deve ser esvaziado
 - A Válvula V0 enche o tanque e a válvula V1 seca o tanque.