INE5430
Inteligência Artificial

Tópico:

- Mecanismos de Raciocínio em Regras de Produção e Algoritmo RETE

Introdução

- Relembrando:
  - As principais características do motor de inferência disponível em shells para sistemas especialistas dizem respeito às seguintes funcionalidades:
    - Método de Raciocínio,
    - Estratégia de Busca,
    - Resolução de Conflito e
    - Representação de Incerteza e Imprecisão.
  - Estas características compõem o Mecanismo de Raciocínio do Sistema Especialista.

Introdução

- Relembrando:
  - As regras de produção formam a abordagem mais antiga para representação de conhecimento em Sistema Especialistas.
  - Continuam sendo uma técnica importante e popular para a construção de solucionadores de problemas baseados fortemente em conhecimento.
  - Frequentemente as regras são uma mistura de conhecimento teórico,de heurísticas derivadas da experiência e de regras para tratar casos estranhos e outras exceções à prática normal.

Introdução

- **Tipos de Sistemas de Produção**
  - Sistemas dedutivos
    - Todas as regras que estão habilitadas podem disparar gerando novas asserções na memória de trabalho
  - Sistemas reativos
    - Deve existir uma forma de resolver os conflitos entre as diferentes regras que estejam habilitadas em cada ciclo

Raciocinando com Encadeamento Progressivo

- Dos dados à conclusão - *data-driven inference*
  - Parte dos fatos na Base de Regras (BR) e na Memória de Trabalho (MT), buscando quais regras eles satisfazem, para produzir assim novas *conclusões* (fatos) e/ou realizar *ações.*
- Três etapas:
  - Busca, Casamento (unificação), Resolução de conflito (para Sistemas Reativos)
- É uma estratégia de inferência muito rápida
  - usada em sistemas de monitoramento e diagnóstico em tempo real.
- Ferramentas comerciais que implementam esta estratégia
  - CLIPS, Jess, Drools, BizTalk, Rules Engine, Soar, OPS5, OPS85, IBM: TIRS

Encadeamento progressivo: algoritmo

1. Armazena as regras da BR na máquina de inferência (MI) e os fatos na memória de trabalho (MT);
2. Adiciona os dados iniciais à memória de trabalho;
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;
5. Dispara a regra selecionada e verifica o seu conseqüente:

   a) se for um fato, atualiza a MT

   b) se for uma ação, chama o procedimento que ativa os efetuadores do agente e atualiza a MT
6. Repete os passos 3, 4 e 5 até que o conjunto de conflito se torne vazio.

Encadeamento progressivo: Busca e Casamento

- Busca e Casamento (unificação)
  - *Unifica as premissas das regras com os fatos da memória de trabalho*
  - ex.: fatos e regra sobre automóveis
    - **MT1:** veloz(Kadet-2.0), veloz(BMW), veloz(Gol-2.0), veloz(Mercedes), importado(BMW), importado(Mercedes)
    - **BR: Se** veloz(x) **e** importado(x) **então** caro(x)
    - **MT2:** MT1 + {caro(BMW), caro(Mercedes)}

Encadeamento progressivo: Busca e Casamento

- Custo da busca-casamento
  - Se a BR é muito grande, verificar todas as premissas de todas as regras a cada ciclo é caro!

Introdução

- Exemplo Inicial:
  - Regra 1: se
    o motor está recebendo combustível e
    o motor tenta pegar
    então
    o problema é vela.
  - Regra 2: se
    o motor não tenta pegar e
    as luzes não acendem
    então
    o problema é bateria ou cabo.
  - Regra 3 se
    o motor não tenta pegar e
    as luzes acendem
    então
    o problema é motor de partida.
  - Regra 4: se
    houver combustível no tanque e
    houver combustível no carburador
    então
    o motor está recebendo combustível.

Heurísticas e Controle de Execução das Regras

- Um método importante para o programador controlar a busca é através da estruturação e ordenação das regras na base de conhecimento.
- Podemos, por exemplo, ordenar as premissas de uma regra de modo que, em primeiro lugar seja testado aquilo que seja mais provável de NÃO ser válido, ou então, o mais fácil de se confirmar.
- Isto possibilita eliminar uma regra (e com isto, uma parte do espaço de busca) o mais cedo possível.
- A regra 1 do exemplo inicial é ineficiente pois ao tentar determinar se o motor está recebendo combustível, invoca outro regra que acaba fazendo duas perguntas ao usuário.

Heurísticas e Controle de Execução das Regras

- Se a ordem das premissas for invertida, uma resposta negativa à consulta "o motor tenta pegar?" exclui esta regra das considerações.
- Também faz mais sentido testar se o motor está tentando pegar antes de verificar se ele está recebendo combustível, pois se ele não tenta pegar, não importa se ele está ou não recebendo combustível.

Heurísticas e Controle de Execução das Regras

- A forma mais simples de realizar a Unificação, procurando na Base de Regras todas as premissas que casem com os fatos da Memória de Trabalho é ineficiente.
- Solução: Algoritmo Rete (rede)
- Vantagens:
  - Elimina duplicações entre regras
  - Elimina duplicações ao longo do tempo

Algortimo Rete

- Inventado por Dr. Charles Forgy em 1979
- Duas partes:
  - Tempo de Compilação
    - Descreve como gerar uma rede de discriminação para as regras da base que possa auxiliar a fase de casamento
    - A rede de discriminação é utilizada com um "filtro de dados"
  - Tempo de Execução
    - A rede é utilizada para unificar a memória de trabalho com as regras da base de forma mais eficiente

Algortimo Rete

- Base de Regras:
  - A(x) ^ B(x) ^ C(y) => add D(x)
  - A(x) ^ B(y) ^ D(x) => add E(x)
  - A(x) ^ B(x) ^ E(x) => delete A(x)
- Memória de Trabalho:
  - {A(1), A(2), B(2), B(3), B(4), C(5)}

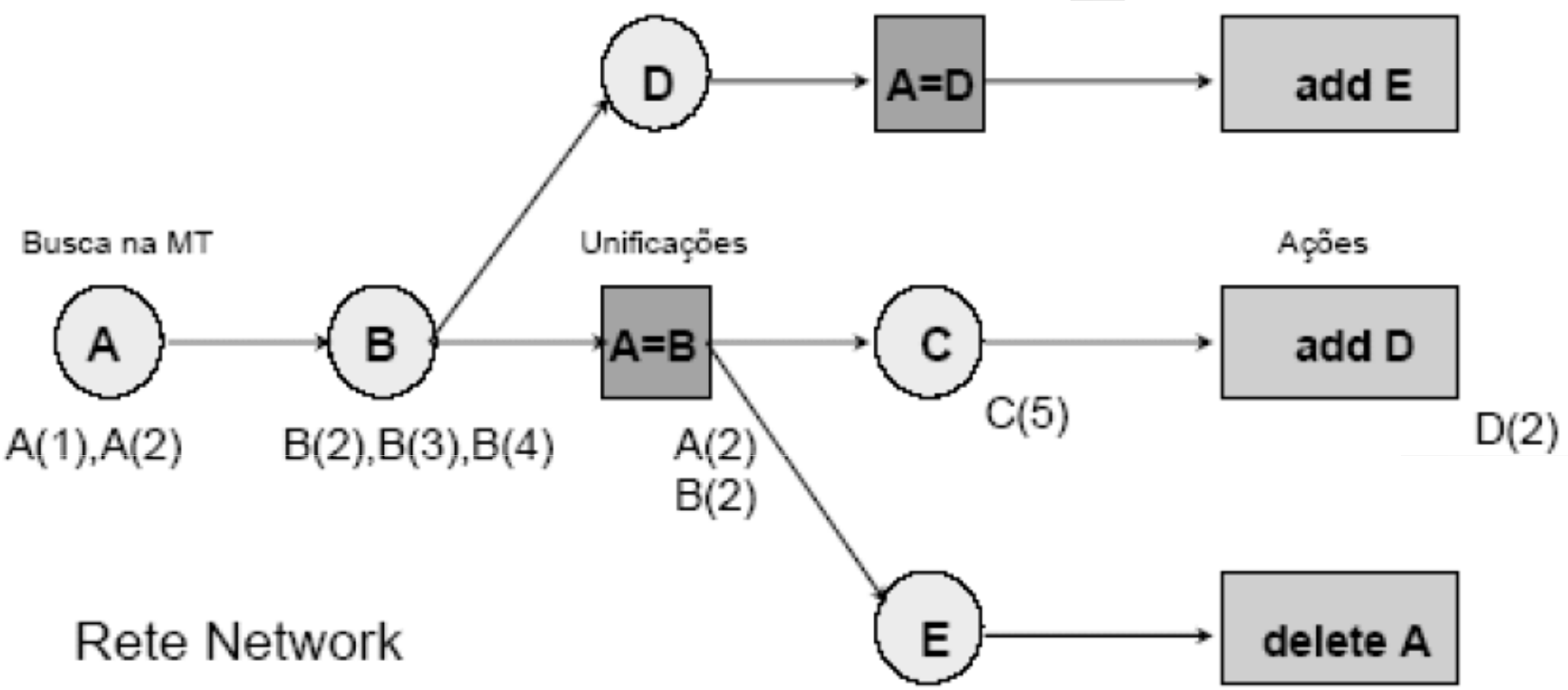
Algortimo Rete

Base de Regras:

A(x) ^ B(x) ^ C(y) => add D(x)
A(x) ^ B(y) ^ D(x) => add E(x)
A(x) ^ B(x) ^ E(x) => delete A(x)

Memória de Trabalho:

{A(1),A(2),B(2),B(3),B(4),C(5)}



Encadeamento Progressivo: resolução de conflitos

- Resolução de Conflitos
  - O sistema decide quais as regras que devem ser executadas
  - Nesta fase podemos utilizar algumas estratégias de controle:
    - Refração: não executar a mesma regra com os mesmos argumentos duas vezes
    - Recência: dar preferência as regras que se referem a elementos da MT criados recentemente (simula o foco de atenção do discurso)
    - Especificidade: dar preferência as regras que são mais específicas
    - Prioridade de Operação: dar preferência as ações com prioridade maior, especificada por alguma critério