

# Sistemas de Produção

---

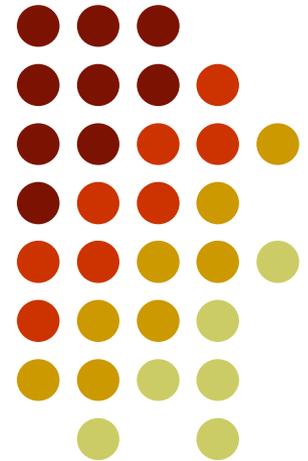
Introdução à Inteligência Artificial

Profa. Josiane

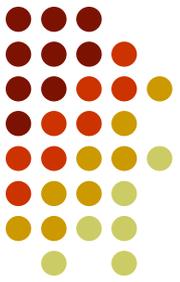
Patrick Henry Winston

*“Artificial Intelligence”* – 3ª edição – cap. 7

agosto/2007

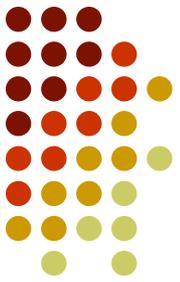


# Regras de Produção



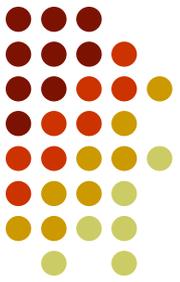
- Inventada em 1943 por Post
- Usa regra formada por pares de condição-ação
  - **Se** condição (ou premissa ou antecedente) ocorre;  
**então** ação (resultado, conclusão ou conseqüente) deverá (ou deveria) ocorrer
- Podem ser vistas como uma simulação do comportamento cognitivo de especialistas humanos
- Representam o conhecimento de forma modular
- Cada regra representa um “pedaço” de conhecimento independente (mas a consistência deve ser mantida)
- Exemplo:
  - **Se** o sinal está vermelho e os carros estão parados  
**então** você pode atravessar a rua

# Sistemas de Produção



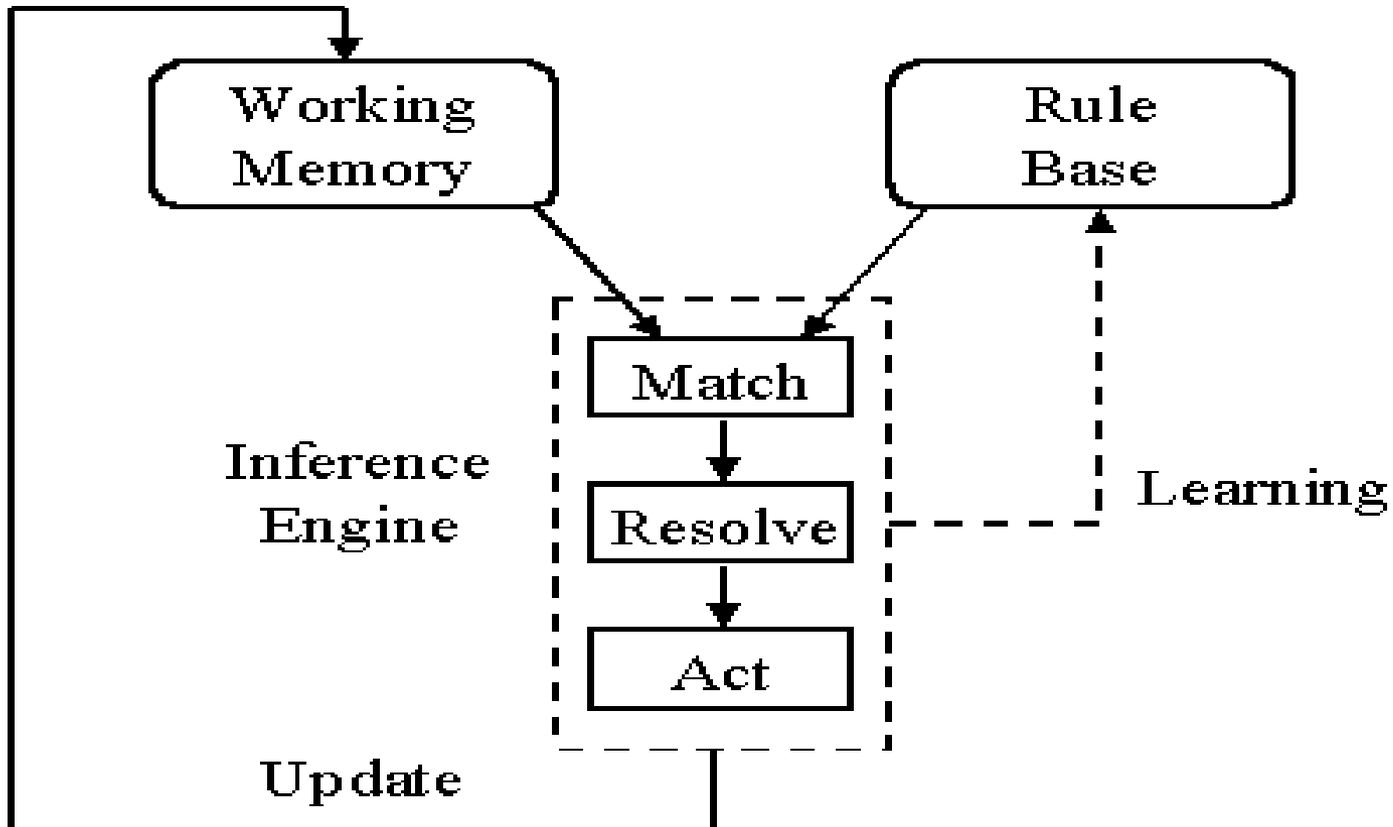
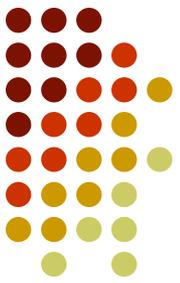
- São sistemas baseados em regras de produção
- Características
  - Separam o conhecimento permanente do conhecimento temporário (base de regras e memória de trabalho)
  - Seus módulos são estruturalmente independentes
  - Sua modularidade facilita a independência funcional
  - Podem utilizar uma variedade de esquemas de controle
- Regras de produção X Máquina de Turing
  - Os sistema de produção têm o mesmo poder de uma máquina de Turing (i.e. podem representar tudo que é computável)

# Sistemas de Produção

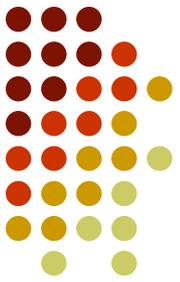


- Componentes:
  - **Memória de Trabalho:** consiste em uma coleção de assertivas verdadeiras (somente literais positivos sem variáveis)
  - **Memória de Regras:** consiste de conjunto de sentenças (regras de inferência) que determinam as ações que devem ser tomadas de acordo com as percepções
    - $p1 \wedge p2 \wedge \dots \Rightarrow act1 \wedge act2 \wedge \dots$
  - **Motor de Inferência:** é a parte do sistema que determina o método de raciocínio, utiliza estratégias de busca e resolve conflitos

# Sistemas de Produção

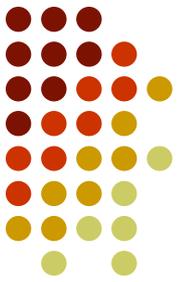


# Tipos de Sistemas de Produção



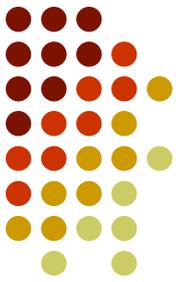
- Sistemas dedutivos
  - Todas as regras que estão habilitadas podem disparar gerando novas asserções na memória de trabalho
- Sistemas reativos
  - Deve existir uma forma de resolver os conflitos entre as diferentes regras que estejam habilitadas em cada ciclo

# Um Sistema Dedutivo que Identifica Animais: ZOOKEEPER



- Características do ZOOKEEPER:
  - Utiliza regras sem antecedentes longos
  - Gera assertivas intermediárias como saídas das regras
  - Combina estas assertivas e aquelas originais para produzir uma conclusão
  - Observa hábitos e características físicas para identificar os animais
  - Neste exemplo, pode distinguir quatro animais: leopardo, tigre, girafa e zebra.

# Regras do ZOOKEEPER



## Regra Z1

Se ?x tem cabelo

Então ?x é um mamífero

- Regra que observa características físicas.

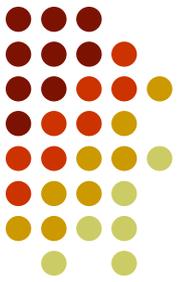
## Regra Z2

Se ?x dá leite

Então ?x é um mamífero

- Regra que determina a classe biológica dos animais, onde Z2 observa hábitos.

# Regras do ZOOKEEPER



## Regra Z3

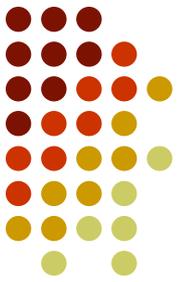
Se            ?x é um mamífero  
              ?x come carne  
Então        ?x é um carnívoro

## Regra Z4

Se            ?x é um mamífero  
              ?x tem dentes pontudos  
              ?x tem garras  
              ?x tem olhos pontiagudos  
Então        ?x é um carnívoro

- As regras determinam se o animal é carnívoro.
- Z3 observa hábitos e Z4 observa características físicas.
- Estas regras já utilizam assertivas geradas por regras anteriormente definidas

# Regras do ZOOKEEPER



## Regra Z5

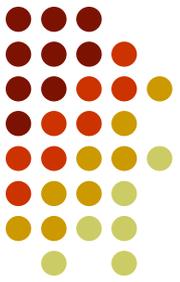
Se           ?x é um mamífero  
              ?x tem cascos  
Então       ?x é um ungulado

## Regra Z6

Se           ?x é um mamífero  
              ?x rumina  
Então       ?x é um ungulado

- As regras determinam se o animal é ungulado
- Z5 observa características físicas e Z6 observa hábitos

# Regras do ZOOKEEPER



## Regra Z7

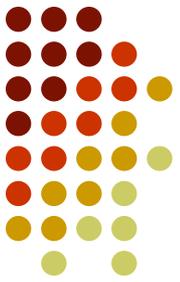
Se           ?x é um carnívoro  
              ?x tem cor amarela tostada  
              ?x tem manchas escuras  
Então       ?x é um leopardo

## Regra Z8

Se           ?x é um carnívoro  
              ?x tem cor amarela tostada  
              ?x tem listas pretas  
Então       ?x é um tigre

- Regras para identificar os animais carnívoros

# Regras do ZOOKEEPER



## Regra Z9

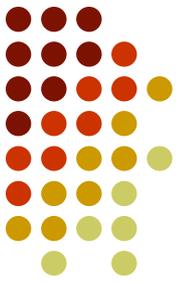
Se           ?x é um ungulado  
              ?x tem pernas longas  
              ?x tem pescoço comprido  
              ?x tem cor amarela tostada  
              ?x tem manchas escuras  
Então       ?x é uma girafa

## Regra Z10

Se           ?x é um ungulado  
              ?x tem cor branca  
              ?x tem listas pretas  
Então       ?x é uma zebra

- Regras para identificar os animais ungulados.

# Encadeamento Progressivo no ZOOKEEPER (Dedutivo)



- Para identificar um animal com o ZOOKEEPER repita:

Para cada regra

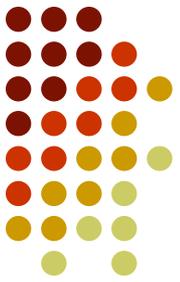
Tente casar cada um dos antecedentes das regras com os fatos conhecidos

Se todos os antecedentes de regras estão casados, execute o conseqüente, a menos que já exista uma assertiva idêntica

Repita para **todas** as alternativas que devem ser testadas

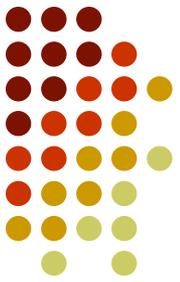
- Até **as regras** não produzirem novas assertivas ou
- Até o animal ser identificado

# Exemplo – Memória de trabalho



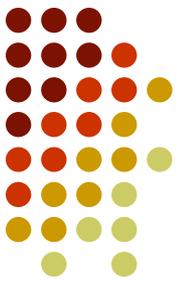
- Catatau tem cabelo
- Catatau rumina
- Catatau tem pernas longas
- Catatau tem cor amarela tostada
- Catatau tem pescoço comprido
- Catatau tem manchas escuras

# Classificar Catatau – Encadeamento para-frente

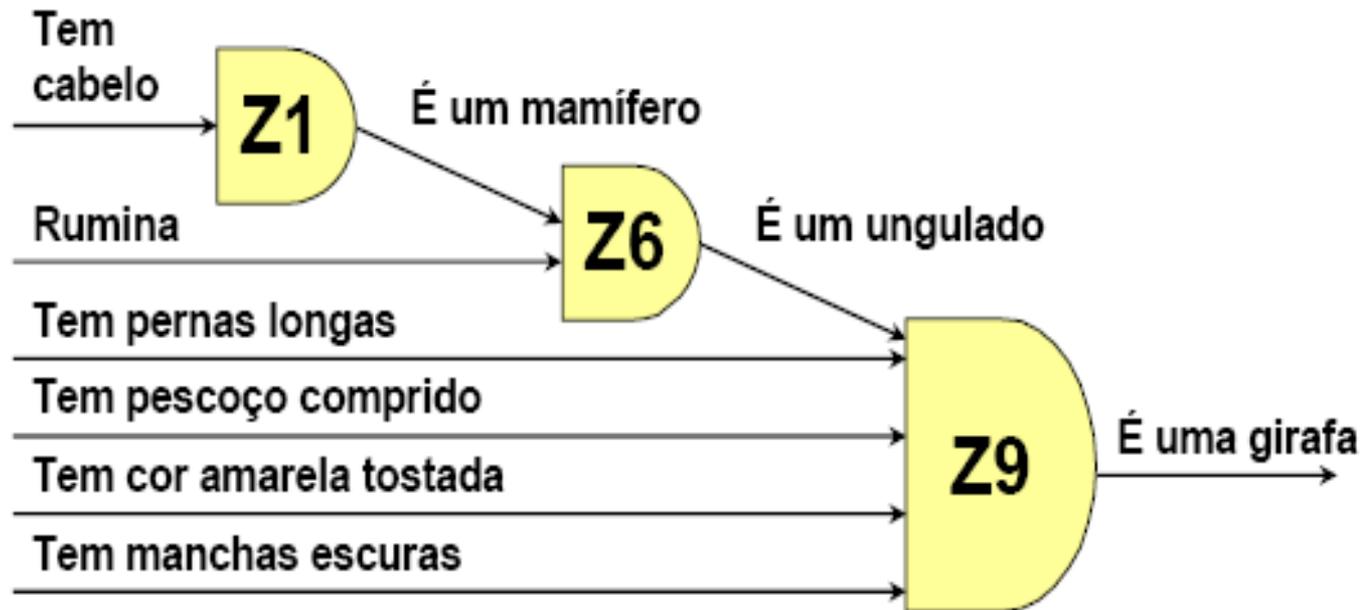


- Verificar que tipo de animal é Catatau
- Solução:
  - Catatau tem cabelo, logo é um mamífero (dispara Z1)
  - Catatau rumina e é um mamífero, logo é um ungulado (dispara Z6)
  - Catatau é um ungulado, tem pernas longas, tem cor amarela tostada, tem pescoço comprido e tem manchas escuras, logo é uma girafa (dispara Z9)

# Encadeamento para-frente

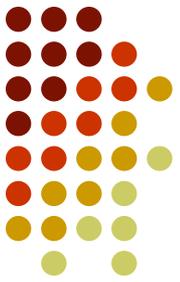


- O fluxo de informações se dá através de uma série de regras antecedente-conseqüente, a partir das assertivas para as conclusões



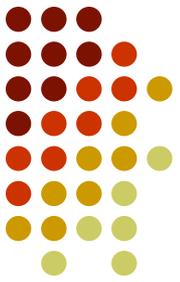
- O encadeamento para-trás é feito tentando provar uma hipótese formulada (conclusão)

# Funcionamento dos Sistemas de Produção



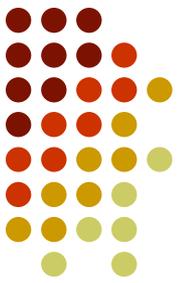
- Três fases:
  - Casamento, **Resolução de conflitos** (p/ sistemas reativos) e Execução
- Casamento
  - O sistema, em cada ciclo, computa o subconjunto de regras cujo as premissas são satisfeitas pelo conteúdo atual da memória de trabalho
  - A forma mais simples de realizar **unificação é ineficiente**, então como solução temos o Algoritmo Rete (rede)
  - Vantagens do Algoritmo Rete:
    - Elimina duplicações entre regras
    - Elimina duplicações ao longo do tempo

# Algoritmo de 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

# Algoritmo de Rete



Base de Regras:

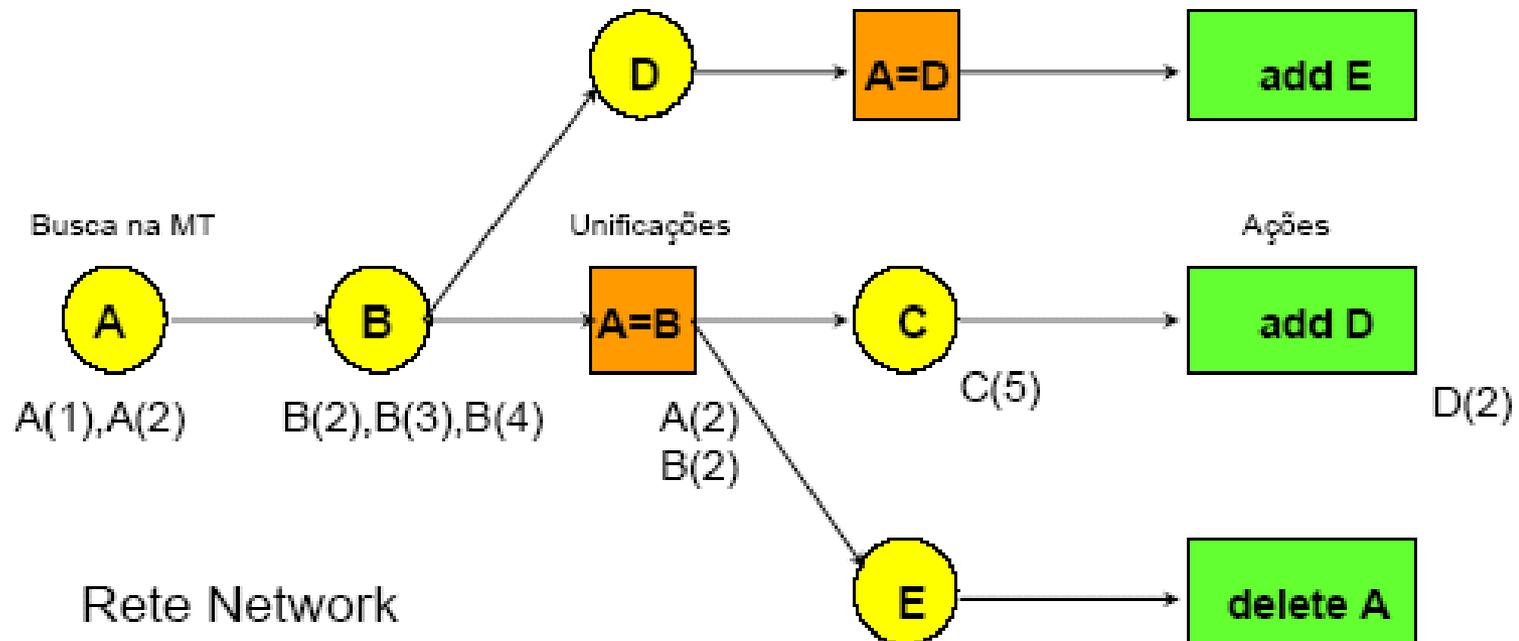
$A(x) \wedge B(x) \wedge C(y) \Rightarrow \text{add } D(x)$

$A(x) \wedge B(y) \wedge D(x) \Rightarrow \text{add } E(x)$

$A(x) \wedge B(x) \wedge E(x) \Rightarrow \text{delete } A(x)$

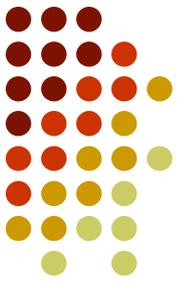
Memória de Trabalho:

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



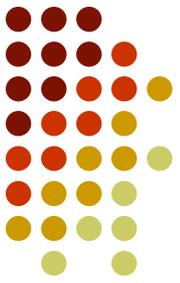
Rete Network

# Funcionamento dos Sistemas de Produção



- 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
- Execução de Ações

# Sistemas de Produção - diagnóstico



- Permitem classificar um item observando suas características
- Exemplo:
  - **P1** luz(fraca)  $\Rightarrow$  bateria(baixa).
  - **P2** luz(ok)  $\Rightarrow$  bateria(ok).
  - **P3** motor(não-gira)  $\wedge$  bateria(baixa)  $\Rightarrow$  assert(bateria-ruim).
  - **P4** motor(não-gira)  $\wedge$  bateria(ok)  $\Rightarrow$  assert(ignição-ruim).
  - **P5** motor(gira)  $\wedge$  medidor\_comb (vazio)  $\Rightarrow$  assert(sem-comb).
- Se todos os fatos estiverem na MT no início as regras apropriadas irão disparar e gerar a conclusão