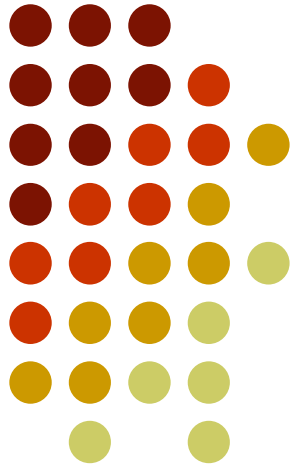
Sistemas de Produção
Sistemas Baseados em Regras

Profa. Josiane

Patrick Henry Winston
"*Artificial Intelligence*" – 3ª edição – cap. 7

agosto/2008



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) ocorre
- 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

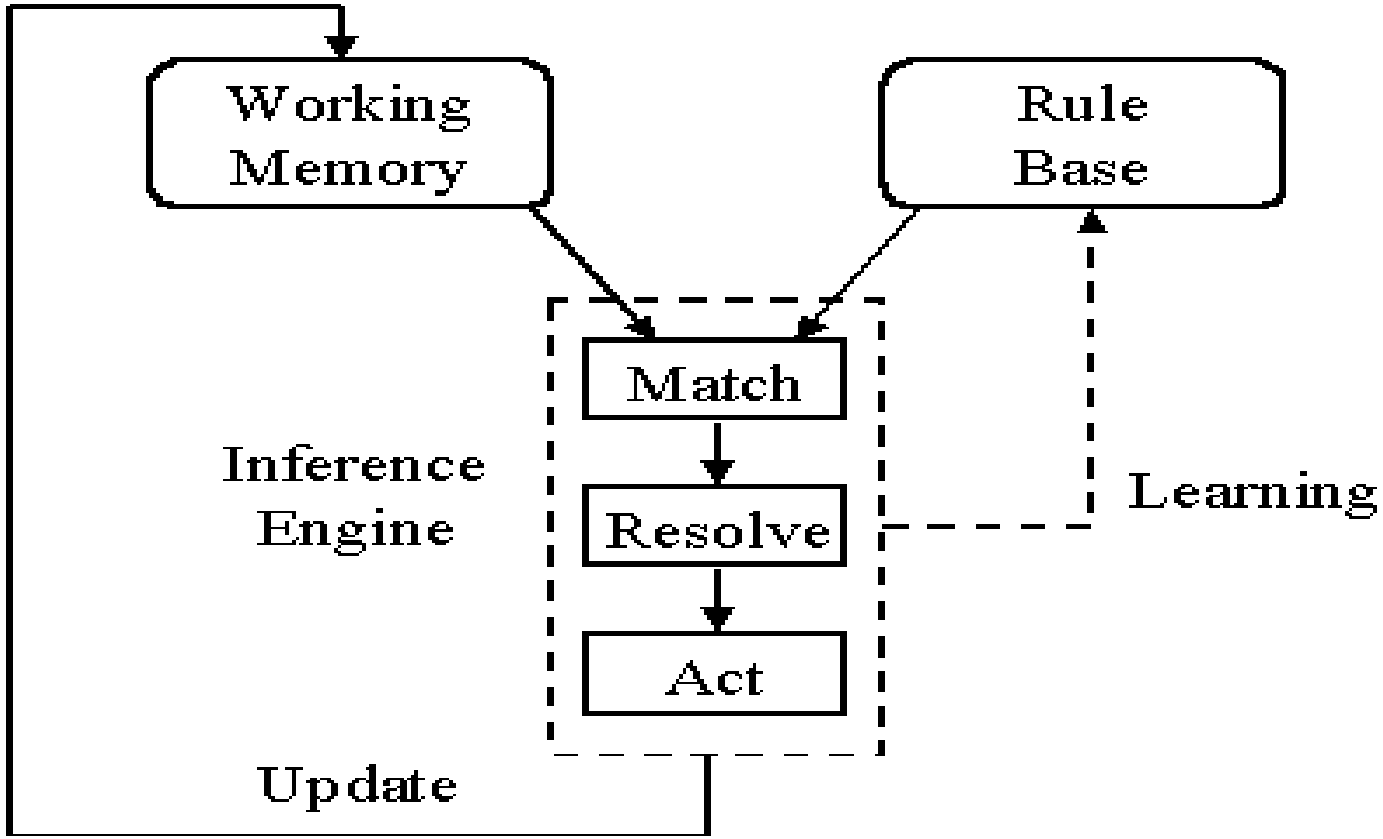
Sistemas de Produção

- 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)

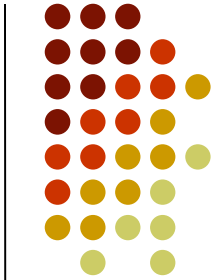
Sistemas de Produção
Componentes

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

Sistemas de Produção
Componentes



Tipos de Sistemas de Produção

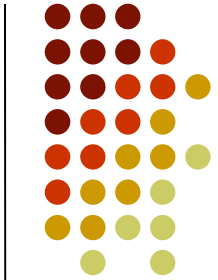


- Sistemas dedutivos
  - Regras: **Se** antecedente **então** consequente
  - Todas as regras que estão habilitadas podem disparar gerando novas asserções na memória de trabalho
- Sistemas reativos
  - Regras: **Se** condição **então** ação
    - Exemplo: **Se** custo < 10 **então** “Coloque o item na sacola”
  - Quando mais de uma regra está habilitada, normalmente somente uma das ações e desejada
    - Deve existir uma forma de resolver os conflitos entre as diferentes regras que estejam habilitadas em cada ciclo

Sistemas de Produção
Funcionamento Geral

- O sistema examina todas os antecedentes das regras baseado na memória de trabalho para determinar o subconjunto de **regras habilitadas**
- O sistema dispara a(s) regra(s) do subconjunto que foram habilitada(s)
  - Em sistemas **dedutivos** todas as regras são disparadas
  - Em sistemas **reativos** este subconjunto é denominado conjunto de conflito e somente uma das regras é disparada
    - A regra disparada é escolhida de acordo com uma **estratégia de resolução de conflitos**
- O(s) consequente(s) da(s) regra(s) são executados
  - Estas ações podem modificar a memória de trabalho, ou a base de regras, executar qualquer tarefa externa ao sistema.
- O loop de disparar regras e executar ações continua até que
  - Nenhuma regra que pode ser habilitada ou
  - Uma ação determine que o programa deve parar

ZOOKEEPER: Um Sistema Dedutivo que Identifica Animais



- 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

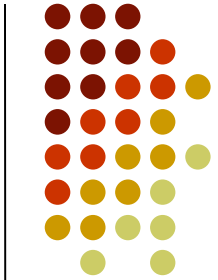
Se	?x é um mamífero
	?x come carne
Então	?x é um carnívoro

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

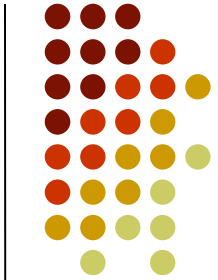
Se	?x é um mamífero
	?x tem cascos
Então	?x é um ungulado

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

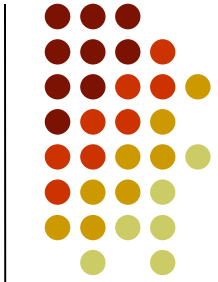
Se	?x é um carnívoro
	?x tem cor amarela tostada
	?x tem manchas escuras
Então	?x é um leopardo

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 para-frente no ZOOKEEPER

- Para identificar um animal com o ZOOKEEPER repita:

Para cada regra

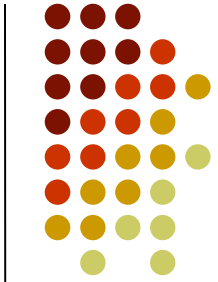
Tente casar cada um dos antecedentes das regras com os fatos conhecidos

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

Repita para **todas** as alternativas de casamento e instanciações

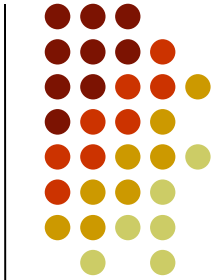
- Até **as regras** não produzirem novas assertivas ou
- Até o animal ser identificado

Memória de trabalho para o ZOOKEEPER: Exemplo



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

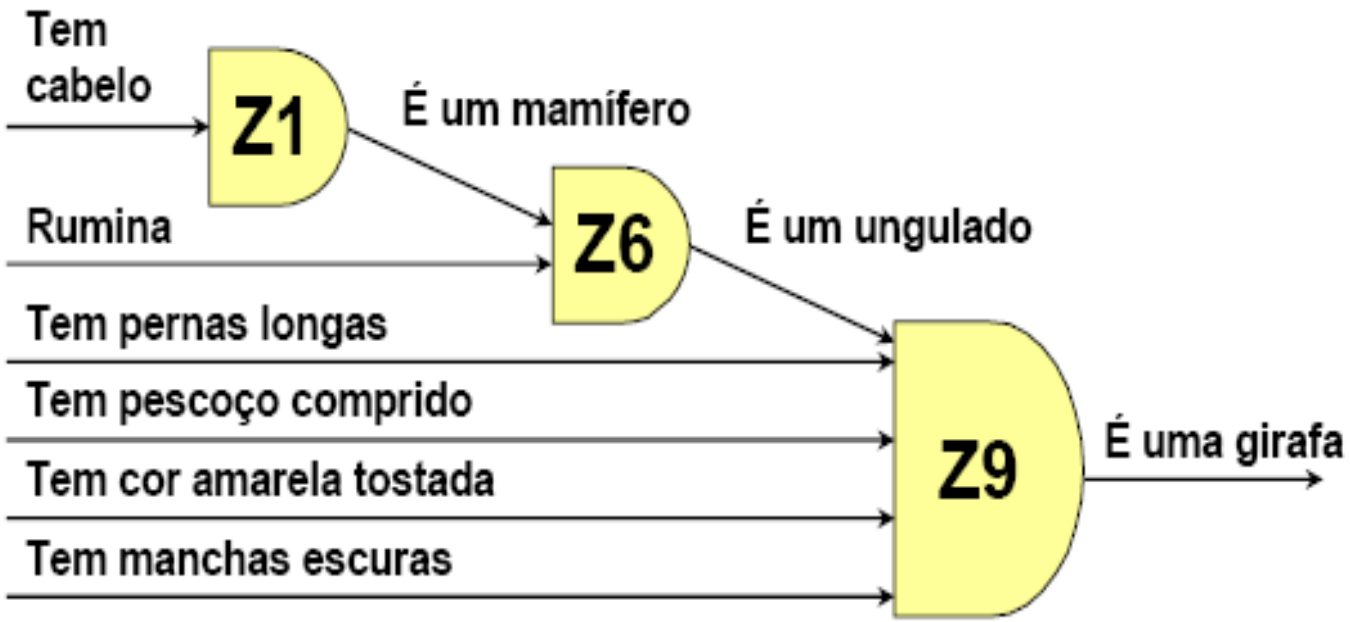
Classificar Catatau – Encadeamento para-frente



- 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



- Exemplo: problema de diagnosticar uma doença com base nos sintomas (memória de trabalho)

Encadeamento para-trás

- Para identificar um animal com o ZOOKEEPER repita:

Para cada hipótese
  Para cada regra que o consequente case com a hipótese corrente
    Tente suportar cada um dos antecedentes da regra
      Com os fatos da memória de trabalho ou
      Por encadeamento para-trás através de outra regra criando uma nova hipótese.
      Tenha certeza de testar todas as alternativas de casamentos e instanciações.
    Se todos os antecedentes da regra são suportados, conclua que a hipótese é verdadeira.

- Até que todas as hipóteses tenham sido testadas e nehuma delas tenha sido suportada ou
- Até que o animal ser identificado.

Encadeamento para-trás

- Devemos formular uma **hipótese** e usar as regras antecedente-conseqüente de trás para frente para suportá-la
- Por exemplo, com a mesma base de regras o ZOOKEEPER pode formular a hipótese *h* = *"Catatau é uma zebra*"
  - Para verificar *h* devemos considerar a regra Z10, que requer que Catatau seja um ungulado, tem cor branca e listas pretas.
  - Devemos checar se Catatau é um ungulado. Como a memória de trabalho não suporta este fato, então formulamos h_1 = *"Catatau é um ungulado".*
  - Podemos utilizar Z5 e Z6 para provar h_1. Utilizando Z5, devemos checar se Catatau é um mamífero e tem cascos.
  - Devemos checar se Catatau é um mamífero. Mas a memória de trabalho não suporta este fato. Então formulamos *h2* = *"Catatau é um mamífero".*

Encadeamento para-trás

- Podemos utilizar Z1 e Z2 para provar h_2. Utilizando Z1, a regra requer que Catatau tenha cabelo.
- De acordo com a memória de trabalho, Catatau tem cabelo, então h_2 é verdadeira. Catatau é um mamífero. Voltamos então para a regra Z5.
- Z5 requer que Catatau seja um mamífero e tenha cascos para ser um ungulado, mas a memória de trabalho não suporta este fato.
- Então, não podemos utilizar Z5 para provar h_1. Vamos tentar Z6, que requer que Catatau seja um mamífero e rumine para ser um ungulado.
- Acabamos de provar h_2 = *"Catatau é um mamífero"*. De acordo com a memória de trabalho, Catatau rumina. Então h_1 é verdadeira. Catatau é um ungulado. Voltamos para a regra Z10.

Encadeamento para-trás

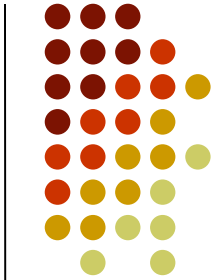
- Z10 requer que Catatau seja um ungulado, tenha cor branca e listas pretas, para ser uma zebra (*h*).
- Os fatos “tem cor branca” e “tem listas pretas” não fazem parte da memória de trabalho e nenhuma regra da base de regras tem este fato como consequente.
- Então concluímos que *h* **não** pode ser provada, e uma nova hipótese deve ser formulada e testada.

- Encadeamento para-trás é utilizado quando um objetivo é especificado e devemos encontrar uma forma de alcançá-lo
  - Ex: Se existe uma epidemia de certa doença, podemos supor que o indivíduo tem a doença e tentar determinar se o diagnóstico está correto baseados nas informações disponíveis.

Qual direção escolher?

- Uma comparação entre o número de antecedentes e consequentes das regras da base pode ajudar a determinar a direção do encadeamento
  - Se, na média, as regras tem mais antecedentes do que consequentes, uma hipótese pode levar a muitas outras hipóteses, então é melhor fazer encadeamento para-frente.
  - Se, ao contrário, as regras tem mais consequentes do que antecedentes, cada fato determina muitos novos fatos ou ações, então é melhor usar encadeamento para-trás
- Se nenhum dos dois é dominante, então o objetivo do sistema pode ajudar na decisão de qual direção usar no encadeamento
  - Se o propósito do sistema é encontrar até onde a informações da memória de trabalho pode nos conduzir, é melhor usar encadeamento para-frente
  - Se o objetivo é encontrar se uma das possíveis conclusões é verdadeira, é melhor usar encadeamento para-trás

Sistemas de Produção Reativos



- As regras são da forma: **Se** condição **então** ação
  - A ação pode inserir uma nova asserção na MT, ou apagar uma asserção existente, ou executar algum outro procedimento.
- Três fases:
  - Casamento, **Resolução de conflitos** e Execução
- Casamento
  - O sistema, em cada ciclo, computa o subconjunto de regras no qual os antecedentes são satisfeitos 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

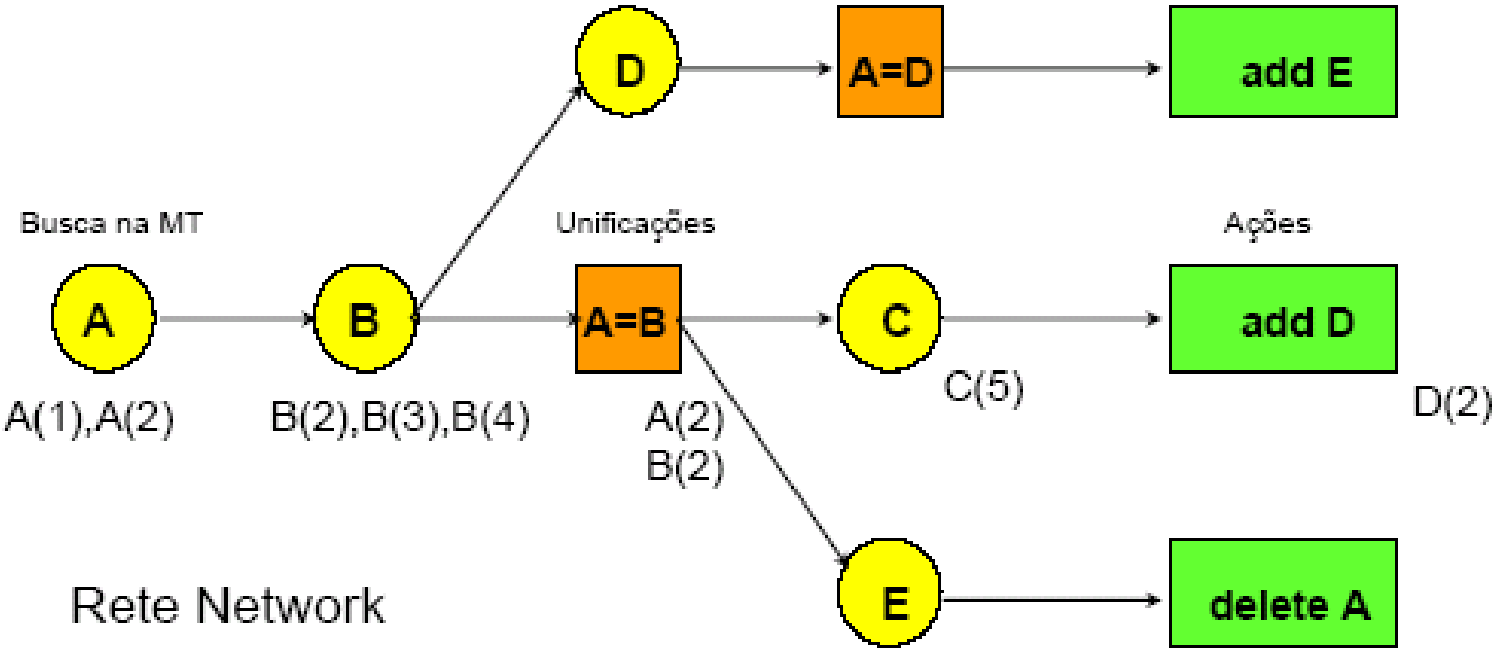
Algoritmo de Rete

Base de Regras:

A(x) ^ 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)}



Sistemas de Produção Reativos

- Estratégias para **resolução de conflitos** entre as regras disparadas
  - **Ordenação das regras**: Ordenar as regras em uma lista de prioridade, executar aquela com maior prioridade
  - **Ordenação de tamanho**: executar a regra tem o maior número de antecedentes
  - **Ordenação de dados**: Ordenar as asserções por prioridade, executar a regra que casa com a asserção de maior prioridade
  - **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
  - Outras

Exercício

- Mostre os quatro (4) primeiros ciclos de operação do sistema de produção a seguir (se possível) listando:
  - i) o Ciclo; ii) as Regras satisfeitas; ii) a Regra escolhida em cada ciclo e iv) a nova memória de trabalho.
- ?x, ?y e ?z são variáveis e os outros termos são constantes.
- A estratégia de resolução de conflitos:
  - Escolher a regra com o maior número de precondições
  - Nenhuma regra pode ser usada mais de uma vez **com o mesmo conjunto de variáveis atribuídas**
  - Se ainda houver conflito entre as regras, a última regra satisfeita deve ser aplicada.
- ¬P(x) significa que P(x) não pode estar na MT

Exercício

- Regras:

  (R1) IF P(?x, ?y) Q(?x, a, ∧ ?z) ∧ ¬Q(?z, a, ?z)
  THEN assert R(?x)
  retract Q(?x, a, ?z)

  (R2) IF R(?x) ∧ Q(?x, ?y, ?z)
  THEN retract R(?x)

  (R3) IF P(f(?x), ?y) ∧ ?x ≤ ?y ∧ Q(f(?w), ?w, f(?z)) ∧ ?y ≤ ?z
  THEN print SUCCESS!

  (R4) IF R(?x)
  THEN assert Q(f(?x), ?x, f(?x))

- Memoria de Trabalho inicial:

  P(f(1), 2) Q(f(1), a, f(1)) R(f(1))
  P(3, f(4)) Q(4, a, 3)
  P(4, 4)