

Inteligência Artificial

Sistemas Baseados em Regras

Paulo Moura Oliveira

Departamento de Engenharias

Gabinete F2.15, ECT-1

UTAD

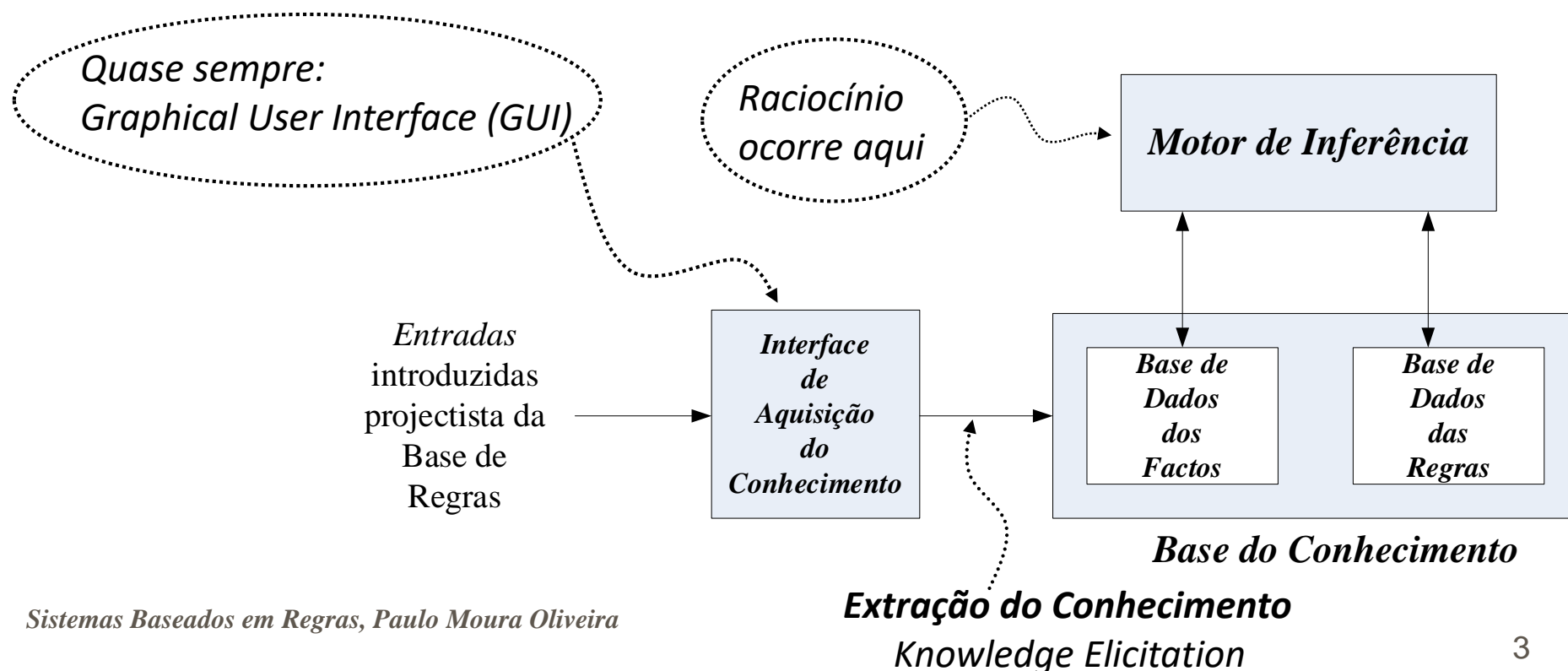
email: oliveira@utad.pt

Nota Preliminar:

- Estes diapositivos enquadram-se na componente de **Representação do Conhecimento e Raciocínio** (*Knowledge Representation and Reasoning*), no contexto da **Inteligência Artificial** e dos **Sistemas Inteligentes**.

Sistemas Baseados em Regras (*Ruled Based Systems* - RBS) são muitas vezes conhecidos também como **Sistemas Baseados no Conhecimento** (*Knowledge Based Systems* - KBS).

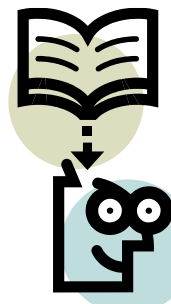
✓ Em que consiste a **construção** de um RBS?



Chama-se **Extração de Conhecimento** ao processo de adquirir conhecimento dos humanos e introduzi-lo nos sistemas ou máquinas

- ✓ A extração do conhecimento pode ser uma tarefa complicada. Aos especialistas nesta área chamam-se:

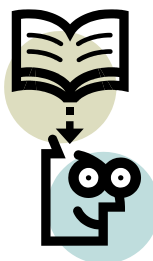
Porque é complicado extrair conhecimento?



*Engenheiros do Conhecimento
(Knowledge Engineers)*

- De uma forma geral o conhecimento dos peritos não é **explícito** mas sim **implícito**, ou seja a pessoa simplesmente sabe fazer uma determinada coisa.

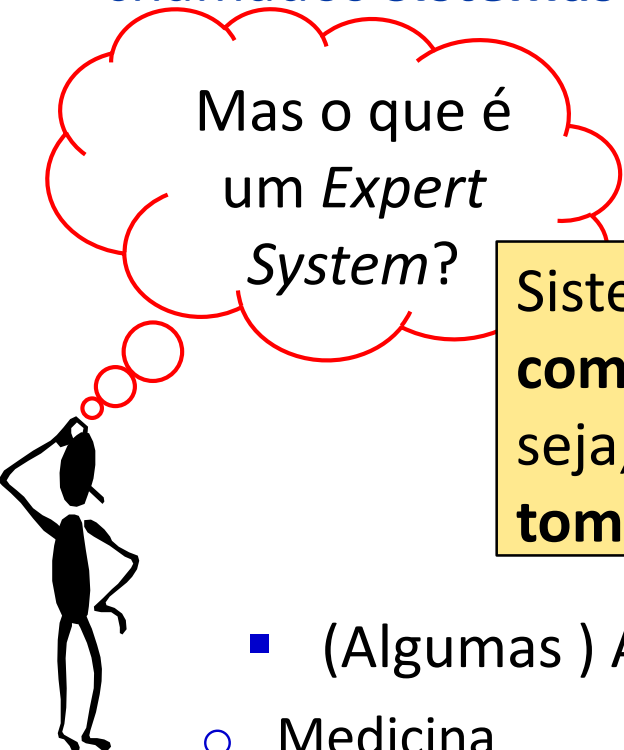
Conhecimento Implícito



Conhecimento Explícito
para que possa ser armazenado e
manipulado num computador

O que é um Sistema Pericial (*Expert System*)?

- ✓ A **extração e representação de conhecimento** é fundamental para os chamados **Sistemas Periciais (*Expert Systems*)** :



Mas o que é
um *Expert*
System?

- Uma tentativa de resposta simples a esta questão é a seguinte:

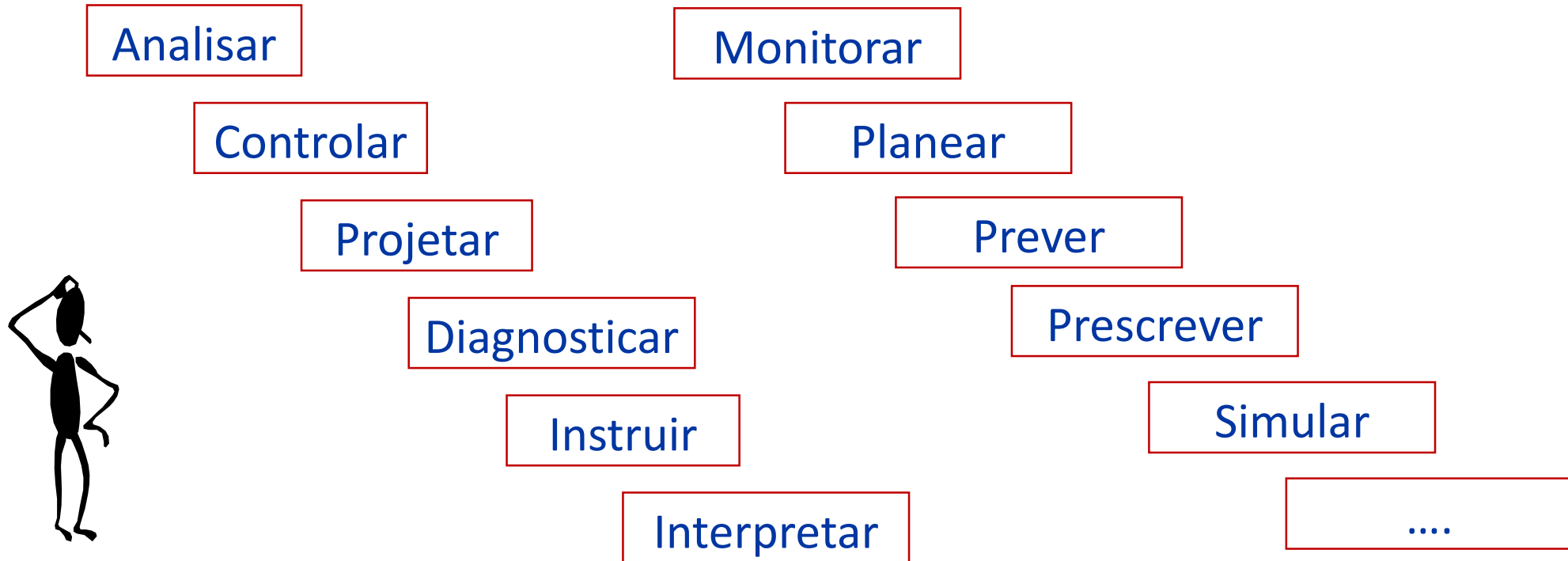
Sistema Pericial (Expert System) – é um **sistema computacional** capaz de emular **peritos humanos**, ou seja, capaz de: **resolver problemas**, de **aprender** e de **tomar decisões**.

- (Algumas) Áreas de aplicação:

- Medicina
- Direito
- Meteorologia
- Processamento de Imagem
- Agricultura
- Engenharia (Controlo, Energia,...)

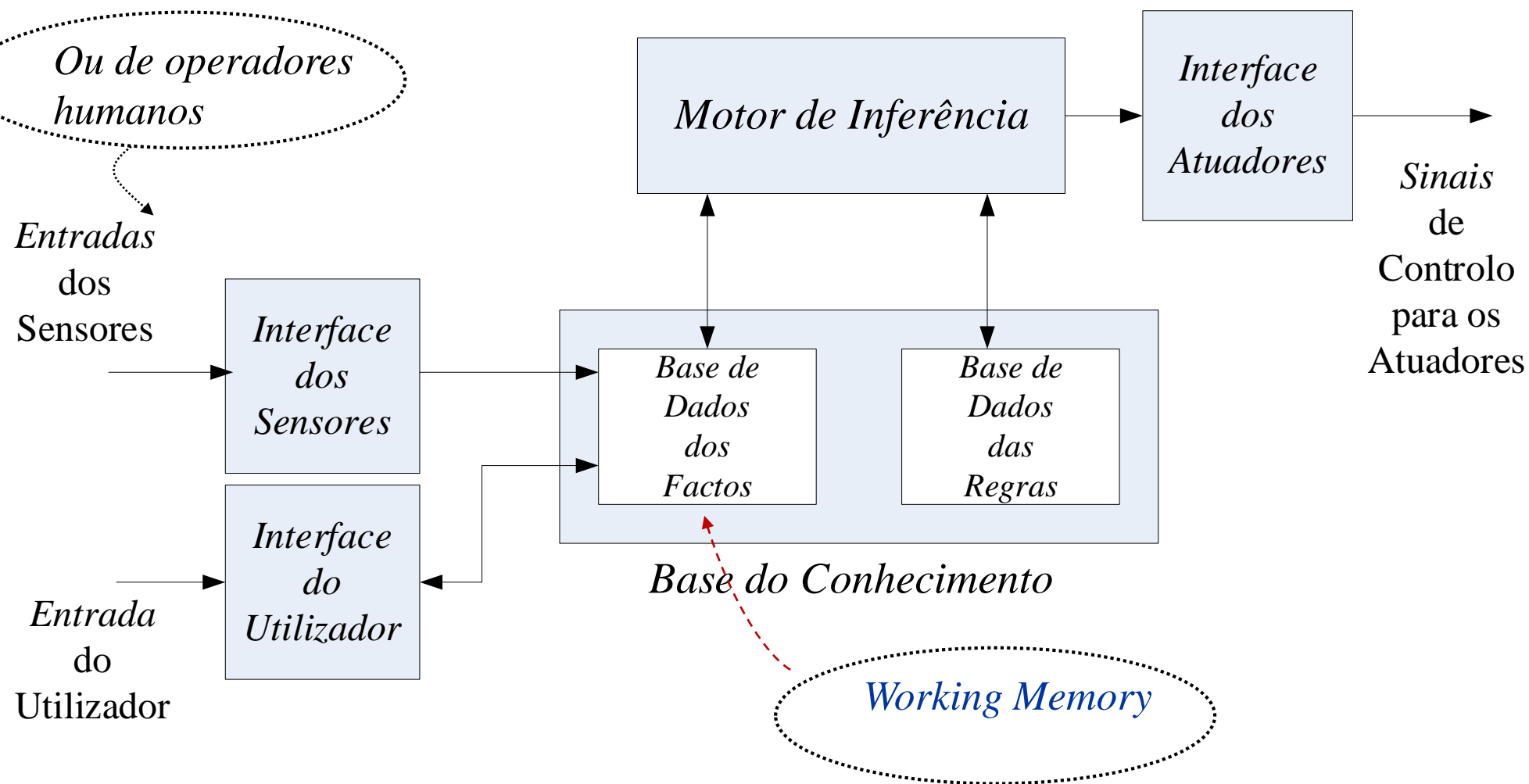
O que é um Sistema Pericial (*Expert System*)?

- ✓ Dependendo da área de aplicação e do propósito dos sistemas periciais, estes podem ser construídos para:



- ✓ Os sistemas baseados em regras baseiam muitos sistemas periciais.

- ✓ Em que consiste a **utilização** de um RBS?





Verdades nalgum
universo relevante!

- A base de conhecimento de um SBR consiste em **factos** e **regras**.

Específicos do problema a que vão ser aplicados.

- A esta especificidade chama-se **domínio** do problema.
- A reivindicação de que alguma coisa é falsa ou verdadeira é chamada **proposição**, ou **predicado**, quando estão envolvidos variáveis.

Exemplo de um predicado:

o aquecedor é ligado para ON é VERDADEIRO

Cláusula $\equiv p$

$T(p) = \text{VERDADEIRO}$

- Os **factos** e as **regras** podem ser divididos em dois grupos:
 1. Conhecimento **Profundo** - relaciona-se com princípios básicos que se assumem que não vão mudar, como por exemplo as leis da física.
 2. Conhecimento **Superficial** - diz respeito a heurísticas que se sabem funcionar por experiência em problemas similares, **mas que podem ser alteradas**.

*O tipo de regras que se mantêm na base de conhecimento tomam geralmente a forma de **regras If-Then**, e são chamadas regras de produção.*

- *Exemplo de regra geral:*

If (alguma coisa é VERDADEIRA)

Then (outra coisa é VERDADEIRA)

- *Exemplo de regra com ação:*

If (alguma coisa é VERDADEIRA)

Then (executar uma ação)

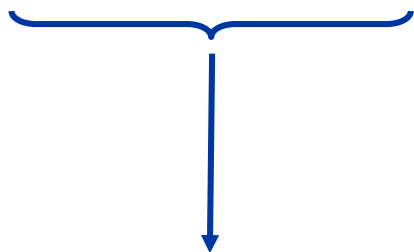
If $T(\text{frio})$

Then $T(\text{aquecimento ON})$

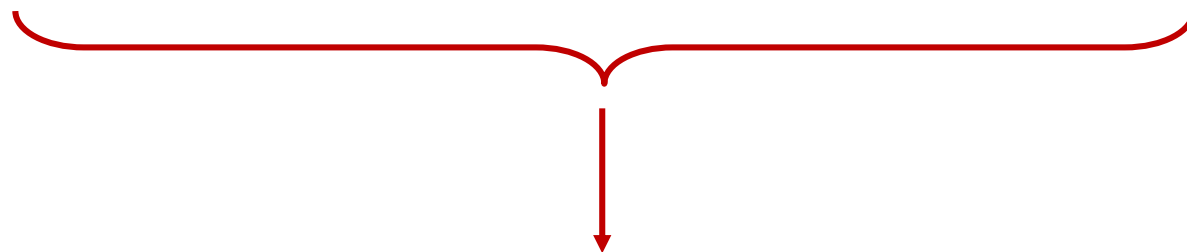
- *Definição do termo frio:*

$\begin{aligned} T(\text{frio}) &= \text{VERDADEIRO se a temperatura dum sensor é menor que } 10^{\circ}\text{C} \\ &= \text{FALSO se a temperatura dum sensor é maior ou igual a } 10^{\circ}\text{C} \end{aligned}$

$T(\text{aquecimento ON}) = \text{VERDADEIRO se o aquecedor está regulado no ON}$
 $= \text{FALSO se o aquecedor está regulado no OFF}$



Antecedente da regra:
Contém uma condição
que tem de ser satisfeita.




Consequente da regra:
Contém a consequência do antecedente ser
VERDADEIRO

Motor de Inferência

Num sistema CBR, uma regra:

- diz-se **engatilhada (triggered)** se o seu antecedente é VERDADEIRO.
- diz-se **disparada** se foi utilizada.
- se não foi disparada diz-se que **falhou**, (antecedente falso ou não selecionada).



Como decidir
qual das regras
engatilhadas se
dispara?

O Motor de Inferência controla quais as regras a disparar, podendo fazê-lo usando várias formas, tais como:

- Encadeamento para a frente
- **Forward Chaining** -
- Encadeamento para trás
- **Backward Chaining** -

Encadeamento para a frente (*Forward Chaining*) – Data Driven

- No *encadeamento para a frente* (*progressiva*) o motor de inferência funciona por *ciclos*.

Em cada *ciclo*:

1. os **factos na memória de trabalho** são atualizados a partir da informação introduzida/deduzida desde o último ciclo
2. as **regras são examinadas** e todas as regras cujos antecedentes são satisfeitos **são engatilhadas**.

*Ao conjunto de regras engatilhadas chama-se **conjunto de conflito**.*

3. O **conflito tem de ser resolvido** de forma a que só uma regra seja **disparada** (nesse ciclo).

O raciocínio começa a partir dos dados disponíveis (guiado por dados) e a memória de trabalho (*working memory*) é comparada com a parte esquerda (antecedente) das regras.

Encadeamento para a frente (*Forward Chaining*)

❑ Exemplo: Sistema de Segurança [1]

Base de dados das Regras

#	
R1	<i>If</i> $T(\text{imagem contém uma cara})$ <i>and</i> $T(\text{cara reconhecida})$ <i>Then</i> $T(\text{abrir porta})$ <i>and</i> $\sim T(\text{imagem contém uma cara})$ <i>and</i> $\sim T(\text{cara reconhecida})$
R2	<i>If</i> $T(\text{imagem contém uma cara})$ <i>and</i> $\sim T(\text{cara reconhecida})$ <i>Then</i> $T(\text{alertar guarda})$ <i>and</i> $\sim T(\text{imagem contém uma cara})$
R3	<i>If</i> $\sim T(\text{imagem contém uma cara})$ <i>Then</i> $\sim T(\text{abrir porta})$ <i>and</i> $\sim T(\text{alertar guarda})$

Base de dados dos Factos:

#	
F1	$\sim T(\text{imagem contém uma cara})$
F2	$\sim T(\text{cara reconhecida})$
F3	$\sim T(\text{abrir porta})$
F4	$\sim T(\text{alertar guarda})$

- Inicialmente, quando o sistema arranca, **só a regra 3 está engatilhada.**

Encadeamento para a Frente (*Forward Chaining*)

❑ Exemplo: Sistema de Segurança (cont.) [1]

- Suponha que um visitante chega à porta, o sistema muda a base de factos para passar a conter $T(\textit{imagem contém uma cara})$, mas o teste de reconhecimento facial falha e o predicado $\sim T(\textit{cara reconhecida})$ mantém-se inalterado na base dos factos.

Base de dados dos Factos:

#	
F1	$T(\textit{imagem contém uma cara})$
F2	$\sim T(\textit{cara reconhecida})$
F3	$\sim T(\textit{abrir porta})$
F4	$\sim T(\textit{alertar guarda})$

Encadeamento para a Frente (*Forward Chaining*)

❑ Exemplo: Sistema de Segurança (cont.) [1]

- Suponha que um visitante chega à porta, o sistema muda a base de factos para passar a conter $T(\text{imagem contém uma cara})$, mas o teste de reconhecimento facial falha e o predicado $\sim T(\text{cara reconhecida})$ mantém-se inalterado na base dos factos.

Regra 1

O 1º antecedente da regra 1 é verificado, mas o 2º não.

Regra 2

Ambos antecedentes da regra são verificados e a regra é engatilhada.

Regra 3

O antecedente da regra não é verificado.

#	
R1	If $T(\text{imagem contém uma cara})$ and $T(\text{cara reconhecida})$ Then $T(\text{abrir porta})$ and $\sim T(\text{imagem contém uma cara})$ and $\sim T(\text{cara reconhecida})$
R2	If $T(\text{imagem contém uma cara})$ and $\sim T(\text{cara reconhecida})$ Then $T(\text{alertar guarda})$ and $\sim T(\text{imagem contém uma cara})$
R3	If $\sim T(\text{imagem contém uma cara})$ Then $\sim T(\text{abrir porta})$ and $\sim T(\text{alertar guarda})$

Encadeamento para a Frente (*Forward Chaining*)

❑ Exemplo: Sistema de Segurança (cont.) [1]

Ciclo	Memória	Conjunto Conflito	Regra Disparada
1	$T(\text{imagem contém uma cara});$ $\sim T(\text{cara reconhecida})$	R2	R2
2			
3			

Encadeamento para a frente (*Forward Chaining*)

❑ Exemplo: Sistema de Segurança (cont.) [1]

- Como a única regra engatilhada é a 2ª, esta é disparada e o predicado $\sim T(\text{ alertar guarda})$ é alterado para $T(\text{ alertar guarda})$ na base de factos.

#	
R2	<i>If</i> $T(\text{ imagem contém uma cara })$ <i>and</i> $\sim T(\text{ cara reconhecida})$ <i>Then</i> $T(\text{ alertar guarda})$ <i>and</i> $\sim T(\text{ imagem contém uma cara })$

#	
F1	$\sim T(\text{ imagem contém uma cara })$
F2	$\sim T(\text{ cara reconhecida})$
F3	$\sim T(\text{ abrir porta})$
F4	$T(\text{ alertar guarda})$

- A regra 2 faz o *reset* de $T(\text{imagem contém cara})$ para $\sim T(\text{imagem contém cara})$ de forma a impedir o seu disparo indefinidamente.
- O sistema de visão vai alterar o facto novamente para $T(\text{imagem contém cara})$ até que o visitante seja admitido por intervenção manual ou se vá embora.

Resolução de Conflitos

- ✓ No caso da existência de um conflito, ou seja mais do que uma regra engatilhada num ciclo é necessário utilizar uma estratégia de resolução de conflitos. Apresentam-se de seguida algumas das mais utilizadas:

Primeira a chegar, primeira a ser atendida (First-come, first-served)

A primeira regra que se determine ter o antecedente satisfeito é disparada.

Ordenação por Antiguidade

A regra que tiver sido disparada há mais tempo é disparada.

Dar prioridades às regras

As regras são ordenadas por prioridades de forma a que a de prioridade mais elevada (decidido pelo projetista) fique em primeiro lugar.

e.g. **Especificidade: Regra com maior número de condições deve ter prioridade.**

Resolução de Conflitos

Dar prioridades aos dados

Os dados ou factos são ordenados por prioridades. A regra que utilize o facto com maior prioridade é disparada.

Ordenação por Especificidade

A regra mais específica (com mais factos verificados) é disparada.

.... etc.

Encadeamento para a frente (*Forward Chaining*)

❑ Exemplo: Previsão do Tempo [2]

- Considere um sistema simplificado para prever o tempo (a partir de um barómetro) baseado no seguinte conjunto de regras e de factos. Pretende-se aplicar ao **encadeamento para a frente**.

#	
R1	<i>If T(ciclone) Then T(nuvens)</i>
R2	<i>If T(anticiclone) Then T(céu limpo)</i>
R3	<i>If T(pressão baixa) Then T(ciclone)</i>
R4	<i>If T(pressão alta) Then T(anticiclone)</i>
R5	<i>If T(seta para baixo) Then T(pressão baixa)</i>
R6	<i>If T(seta para cima) Then T(pressão alta)</i>

#	
F1	<i>T(seta para cima)</i>

Encadeamento para a frente (*Forward Chaining*)

❑ Exemplo: Previsão do Tempo (cont.) [2]

Ciclo	<i>Memória Trabalho</i>	<i>Conjunto Conflito</i>	<i>Regra Disparada</i>
1	<i>T(seta para cima)</i>	<i>R6</i>	<i>R6</i>
2	<i>T(seta para cima); T(pressão alta)</i>	<i>R6, R4</i>	<i>R4</i>
3	<i>T(seta para cima); T(pressão alta); T(anticiclone)</i>	<i>R6, R4,R2</i>	<i>R2</i>
4	<i>T(seta para cima); T(pressão alta); T(anticiclone); T(céu limpo)</i>	<i>R6, R4,R2</i>	<i>Paragem</i>

Encadeamento para trás (*Backward Chaining*) – Goal Driven

- Na cadeia para trás (regressiva), **parte-se do objetivo** (*goal*) retrocedendo para verificar como este pode ser atingido. Este processo chama-se **raciocínio a partir de objetivos**.

O raciocínio começa a partir do estado objetivo (guiado por objetivos) e a memória de trabalho (*working memory*) é comparada com a **parte direita** das regras (consequente).

Encadeamento para trás (*Backward Chaining*) – Goal Driven

❑ Exemplo: Previsão do Tempo [2]

- Considere um sistema simplificado para prever o tempo (a partir de um barómetro) baseado no seguinte conjunto de regras e de factos. Pretende-se aplicar ao **encadeamento para trás**.

#	
R1	<i>If T(ciclone) Then T(nuvens)</i>
R2	<i>If T(anticiclone) Then T(céu limpo)</i>
R3	<i>If T(pressão baixa) Then T(ciclone)</i>
R4	<i>If T(pressão alta) Then T(anticiclone)</i>
R5	<i>If T(seta para baixo) Then T(pressão baixa)</i>
R6	<i>If T(seta para cima) Then T(pressão alta)</i>

#	
F1	<i>T(céu limpo)</i>

Encadeamento para trás (*Backward Chaining*) – Goal Driven

❑ Exemplo: Previsão do Tempo [2]

Ciclo	<i>Memória Trabalho</i>	<i>Conjunto Conflito</i>	<i>Regra Disparada</i>
1	<i>T(céu limpo)</i>	<i>R2</i>	<i>R2</i>
2	<i>T(céu limpo); T(anticiclone)</i>	<i>R2,R4</i>	<i>R4</i>
3	<i>T(céu limpo); T(anticiclone); T(pressão alta)</i>	<i>R2,R4,R6</i>	<i>R6</i>
4	<i>T(céu limpo); T(anticiclone); T(pressão alta); T(seta para cima)</i>	<i>R2,R4,R6</i>	<i>Paragem</i>

Exemplo: Classificação de animais

- Considere um sistema simplificado classificar animais baseado no seguinte conjunto de regras e de factos. Pretende-se aplicar ao encadeamento: i) para a frente e ii) para trás.

#	
R1	<i>If T(animal tem penas) Then T(animal é um pássaro)</i>
R2	<i>If T(animal voa) and T (animal põe ovos) Then T(animal é um pássaro)</i>
R3	<i>If T(animal é um pássaro) and T (animal é voador exímio) Then T(animal é um albatroz)</i>

#	
F1	<i>T(animal voa)</i>
F2	<i>T(animal tem penas)</i>
F3	<i>T(animal é voador exímio)</i>

Exemplo: Classificação de animais

- Considere um sistema simplificado classificar animais baseado no seguinte conjunto de regras e de factos. Pretende-se aplicar ao encadeamento: i) para a frente e ii) para trás.

i)

Ciclo	Memória Trabalho	Conjunto Conflito	Regra Disparada
1	$T(\text{animal voa}); T(\text{animal tem penas}); T(\text{animal é voador exímio});$	$R1$	$R1$
2	$T(\text{animal voa}); T(\text{animal tem penas}); T(\text{animal é voador exímio});$ $T(\text{animal é um pássaro})$	$R1, R3$	$R3$
3	$T(\text{animal voa}); T(\text{animal tem penas}); T(\text{animal é voador exímio});$ $T(\text{animal é um pássaro}); T(\text{animal é um albatroz})$	$R1, R3$	Paragem

Exemplo: Classificação de animais

ii) O Animal é um Albatroz

Ciclo	Memória Trabalho	Conjunto Conflito	Regra Disparada
1	<i>T(animal é um albatroz)</i>	<i>R3</i>	<i>R3</i>
2	<i>T(animal é um albatroz); T(animal é um pássaro); T(animal é voador exímio);</i>	<i>R3,R1</i>	<i>R1</i>
3	<i>T(animal é um albatroz); T(animal é um pássaro); T(animal é voador exímio); T(animal tem penas)</i>	<i>R3,R1,R2</i>	<i>R2</i>
4	<i>T(animal é um albatroz); T(animal é um pássaro); T(animal é voador exímio); T(animal tem penas); T(animal põe ovos)</i>	<i>R3,R1,R2</i>	<i>Paragem</i>