

# IA

## Agentes

**Agente** - entidade com capacidade de obter informação sobre o seu ambiente

- usa sensores para perceber o ambiente
- usa atuadores para interagir com o ambiente

**Teste de Turing** - AI conversa com um humano. Se o humano não conseguir distinguir entre um AI e outro humano, o AI é "inteligente"

**Mundo:**

- **Acessível** -> agente consegue recolher toda a informação do mundo
- **Efetivamente acessível** -> agente consegue recolher a informação que necessita para uma ação
- **Determinístico** -> se o estado resultante de uma ação for TOTALMENTE determinado pelo estado atual e pelos efeitos esperados dessa ação
- **Episódico** -> se cada percepção-ação for independente das outras
- **Dinâmico/Estático** -> se o mundo **pode/não pode** mudar enquanto o agente pensa
- **Contínuo/Discreto** -> se a evolução do mundo é **contínua / sem saltos / com saltos**

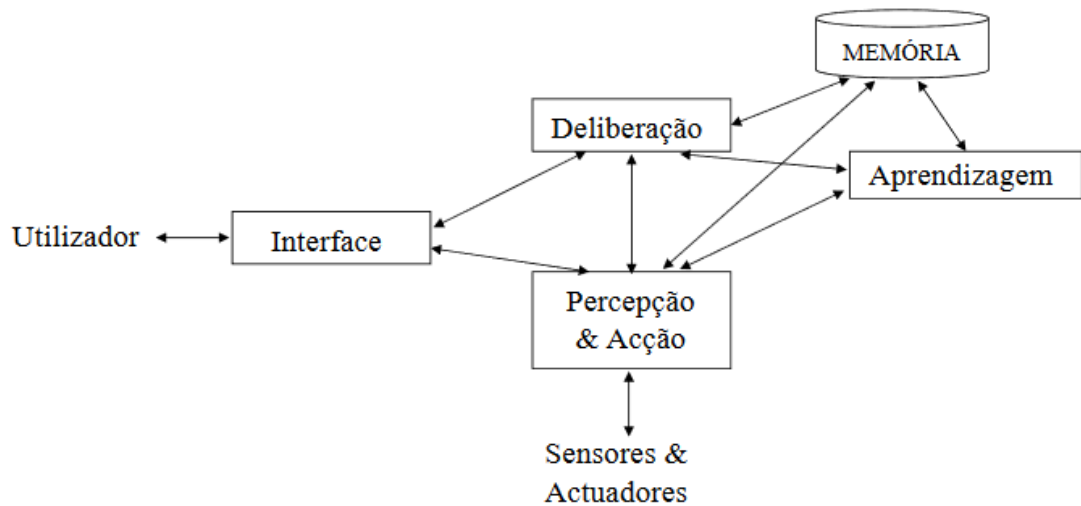
**Agentes reativos** - reagem de acordo com as alterações imediatas do mundo, sem considerar histórico de ações (de ele próprio)

**Agentes deliberativos** - tem modelo interno do mundo, permitindo planeamento de ações, e consideram estados passados e antecipam futuras consequências

**Arquiteturas de agentes:**

- Subsunção: Por camadas. Camada mais baixa é a mais reactiva. Camadas de cima dão "feed" para as camadas de baixo
- Três Torres: Percepção -> Modelo -> Ação
- Três Camadas: Deliberativa -> Ações Intermedias -> Reactiva

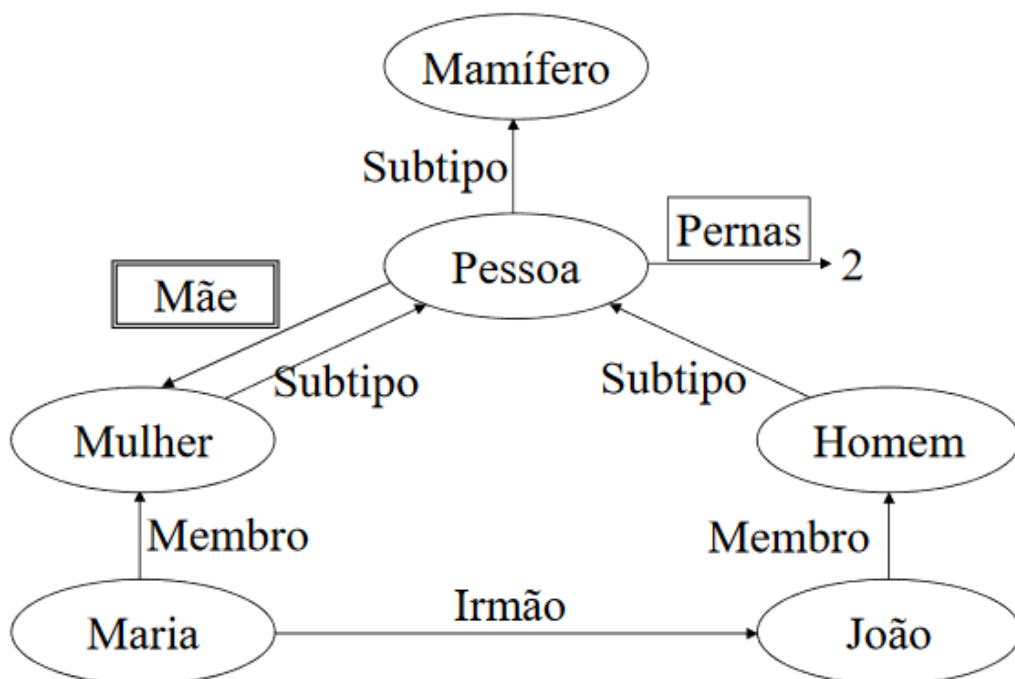
- CARL (jacobs do mr beast):



## Redes Semânticas

### Tipos de Relações:

- Subtipo: mamífero
- Membro
- Relação objecto-objecto
- Relação conjunto-objecto
- Relação conjunto-conjunto



### Herança:

- Apenas em relações "subtipo" e "membro"

### Metodos e Demonios:

- Métodos: semântica parecida com POO
- Demônios: procedimentos que são executados automaticamente quando certas operações (leitura ou escrita) são realizadas

**Dedução:** permite inferir casos particulares a partir de regras gerais

**Indução:** oposto de Dedução. Permite inferir regras gerais a partir de casos particulares

## Linguagem KIF (KYS)

- Feita para representar conhecimento trocado entre agentes
- Pode também ser usada para modelos internos de um agente
- Semântica declarativa
- Tao ou mais expressiva que logica de 1ª ordem
- Permite representar meta-conhecimento
- Mundo é feito com objetos e relações entre objetos
- Relação = conjunto arbitrário de listas de objetos
- Objetos podem ser concretos ou abstratos
- Objetos podem ser primitivos ou compostos

## Engenharia de Conhecimento

**Base de conhecimento:** Conjunto de representações de factos e regras de funcionamento

**Engenharia de Conhecimento:** Processo ou actividade de construir bases de conhecimento

**Ontologia:**

- Vocabulário sobre um domínio com relações hierárquicas
- Objectivo é captar a essência da organização do conhecimento

## Redes de Bayes

- Permitem representar conhecimento impreciso
- Grafo dirigido acíclico

## Pesquisa com propagação de restrições

**Problemas de atribuição:** Pretende atribuir valores a um conjunto de variáveis

**Métodos de resolução:**

- Construtivo
- Construtivo com propagação de restrições
- Melhorias sucessivas

$$\begin{array}{r} \text{T W O} \\ + \text{T W O} \\ \hline \text{F O U R} \end{array}$$

Objectivo é dar um numero entre 0 e 9 a todas as letras de modo a que a soma seja true

F, O, R, T, U, W  $\in \{0, \dots, 9\}$  - Restringir sobre todas as variaveis

Agora restringimos para:

- $O+O = R + 10 \times \text{CarryOver}$  (restrição sobre 3 variaveis)
- $W+W+\text{CarryIn} = U + 10 \times \text{CarryOver}$
- Etc. para as outras colunas

## Pesquisa por melhorias sucessivas

### Montanhismo:

- É visto como um problema de otimizar uma função
- Escolhe sempre a child com **melhor avaliação**
- No backtracking
- if (current\_node.evaluation > child\_node.evaluation) then STOP (atingiu um máximo local)

### Variantes:

- Estocastico: escolhe **aleatoriamente** entre as children que melhoram a avaliação
- Primeira Escolha: escolhe **aleatoriamente** entre as children **ATÉ** encontrar uma com melhor avaliação
- Reinicio Aleatório: montanhismo normal (realizado várias vezes) mas com start\_nodes aleatórios e escolhe a melhor solução

### Recozimento Simulado:

- Aceita children com values **PIORES** que o atual
- Children escolhidos aleatoriamente
- Usa temperatura para terminar a pesquisa:
  - À medida que o tempo passa, a pesquisa arrisca cada vez menos quanto a aceitar alterações com ganho negativo

### Pesquisa local alargada:

- Semelhante ao montanhismo, mas parecido com machine learning. Guarda estados anteriores, e à medida que pesquisa, vai descartando os estados com pior avaliação, e guardando os melhores children

### Pesquisa alargada estocastica:

- Semelhante ao montanhismo, mas parecido com machine learning. Guarda estados anteriores, e à medida que pesquisa, vai guardando os melhores children de forma aleatória

### Algoritmos geneticos:

- Variante da pesquisa alargada estocástica em que os sucessores são gerados por combinação de dois estados, e não apenas por modificação de um único estado

### Aprendizagem:

- Supervisionada: cada exemplo contem uma instancia do que quer aprender
- Semi-supervisionada: uma pequena parte dos exemplos contem uma instancia do que quer aprender
- Por reforço: o agente aprende o seu comportamento tendo em conta as recompensas que recebe pelas suas ações
- Nao-supervisionada: o processo em si descobre o que quer aprender

### Aprendizagem de regras - pesquisa em profundidade ( greedy ):

- Dado um grande numero de exemplos, quando eu fizer um novo teste, ira descartar as partes Negativas, e tentará maximizar as Positivas.

### Árvore de decisão: selecção do atributo de teste

- Baseado na Teoria da Informação:
  - Entropia apriori:  $H(C) = -\sum p(C_i) \times \log_2(p(C_i))$
  - Entropia aposteriori, dado o valor de um atributo:  $H(C|a_{j,k}) = -\sum p(C_i | a_{j,k}) \times \log_2(p(C_i | a_{j,k}))$
  - Entropia global aposteriori:  $H(C|A_j) = \sum p(a_{j,k}) \times H(C|a_{j,k})$
- Ganho de informação:
  - $I(C;A_j) = H(C) - H(C|A_j)$
- Razão do ganho:
  - $H(A_j) = -\sum p(a_{j,k}) \times \log_2(p(a_{j,k}))$
  - $R(C;A_j) = I(C;A_j) / H(A_j)$