

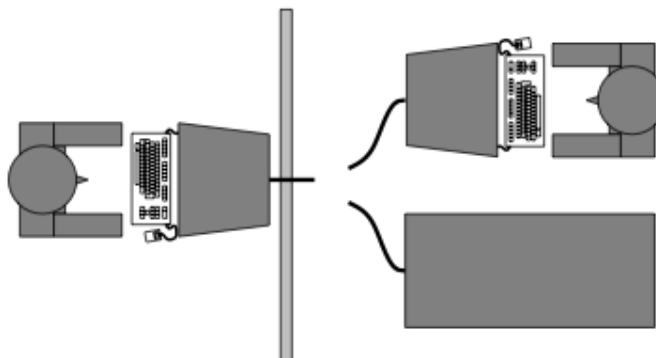
INTELIGÊNCIA ARTIFICIAL

- **INTRODUÇÃO**

- Resolução de problemas
 - Algoritmos tradicionais/"lineares" podem ter altos custos computacionais
 - O conhecimento do problema deve ser usado de forma inteligente
- Dimensões/Abordagens de IA



- Sistemas que atuam como um ser humano
 - Definições
 - “a arte de criar máquinas que realizem actividades que requerem inteligência quando realizadas por pessoas” (Kurzweil, 1990)
 - “como fazer os computadores fazerem coisas nas quais os seres humanos hoje em dia são mais eficientes.” (Rich and Knight, 1991)
 - Teste de Turing
 - Humano interroga um computador
 - Computador passa no teste se o Humano não consegue dizer se existe um computador ou um humano do outro lado

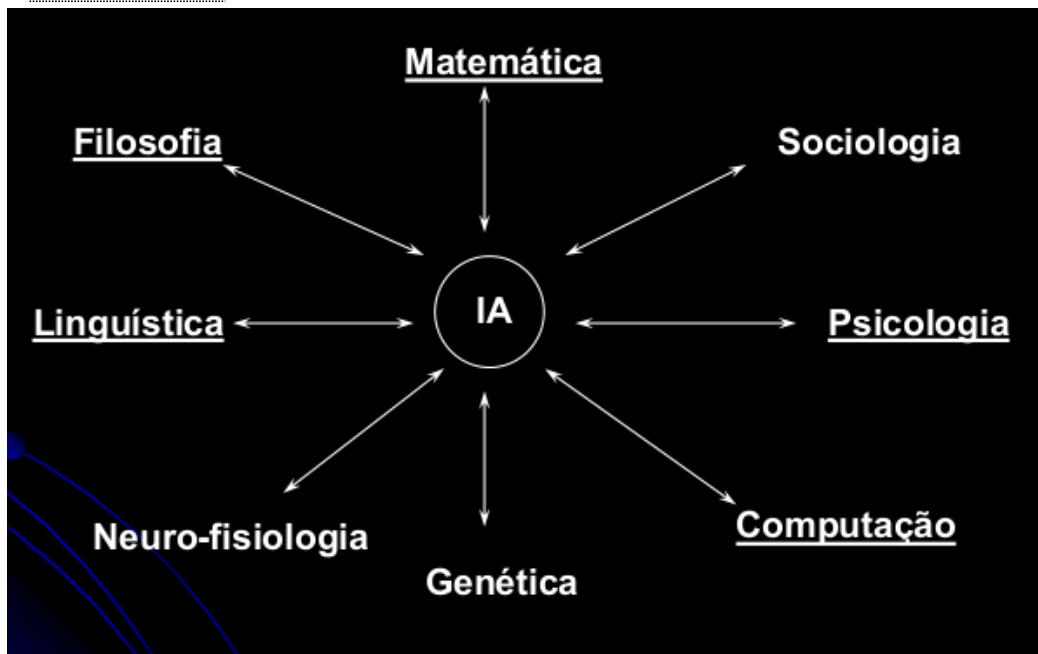


- Turing define inteligência como “a habilidade de obter uma performance de nível humano em todas as tarefas cognitivas de forma a enganar um interrogador humano”

- Sistemas que pensam como um ser humano
 - Definições
 - “O excitante esforço para fazer computadores pensarem, máquinas com mentes, no sentido completo e literal” (Haugeland, 1985)
 - “A automação de actividades que associamos com o pensamento humano, tais como tomada de decisões, solução de problemas e aprendizagem” (Bellman, 1978)
 - Formas de pensamento
 - Introspecção
 - Ciências Cognitivas
 - Neurociências
 - Psicologia experimental
- Sistemas que pensam racionalmente
 - Definições
 - “O estudo das faculdades mentais através do uso de modelos computacionais” (Charniak and McDermott, 1985).
 - “O estudo das computações que fazem possível perceber, raciocinar e agir”(Winston, 1992).
 - Processos de raciocínio
 - Lógica (ex.: lógica dedutiva)
 - Silogismos (Aristóteles)
 - ...
- Sistemas que atuam racionalmente
 - Definições
 - “Um campo de estudo que procura explicar e emular o comportamento inteligente em termos de processos computacionais” (Schalkoff, 1990).
 - “O ramo da ciência de computação que está preocupada com a automação do comportamento inteligente” (Luger and Stubblefield, 1993).
 - Comportamento racional → Cumprir os objectivos a partir das informações disponíveis
 - Agente → algo/alguém que percebe e age
 - Abordagem racional → ênfase às inferências corretas
 - Existem formas de agir racionalmente sem envolver inferências (ex.: tirar a mão de uma panela quente, piscar os olhos quando alguém passa a mão à frente dos nossos olhos)
- Paradigmas de IA
 - Simbólico → metáfora linguística (ex.: sistemas periciais, agentes)
 - Etapas do processo
 - 1) Identificar o conhecimento do domínio (modelo do problema)
 - 2) Representá-lo utilizando uma linguagem formal de representação
 - 3) Implementar um mecanismo de inferência para utilizar esse conhecimento
 - Conexionista → metáfora cerebral (ex.: redes neuronais)
 - Definição “romântica”
 - Técnica inspirada no funcionamento do cérebro, onde neurónios artificiais, conectados em rede, são capazes de aprender e de

generalizar

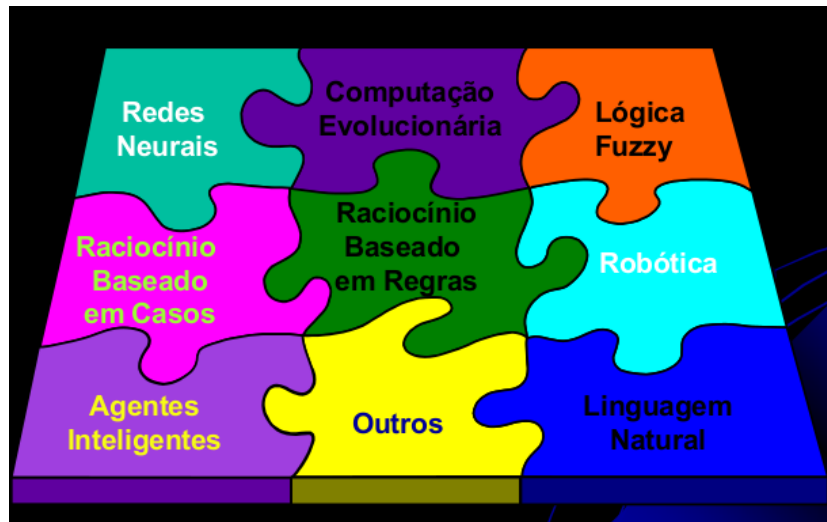
- Definição “matemática”
 - Técnica de aproximação de funções por regressão não linear
- Linguagem → redes de elementos simples
- Raciocínio → aprender directamente a função entrada-saída
- Evolucionista → metáfora da natureza (ex.: algoritmos genéticos, vida artificial)
 - Conceito de evolução
 - Diversidade é gerada por cruzamento e mutações
 - Seleção natural → os seres mais adaptados aos seus ambientes sobrevivem
 - Características genéticas são herdadas pelas próximas gerações
 - Método probabilista de busca para resolução de problemas (otimização) “inspirado” na teoria da evolução
 - Evolui e adapta-se por cruzamento através de sucessivas gerações
- Estatístico/Probabilístico/Possibilístico (ex.: Redes Bayesianas, sistemas difusos)
 - Probabilidades → Raciocínio com incerteza
 - Possibilidades → Raciocínio com imprecisão
- Fundamentos



- Filosofia
 - Sistema de silogismos (Aristóteles)
 - Características
 - Raciocínio organizado
 - Mecanização do processo de geração de conclusão a partir de premissas verdadeiras
 - Regras para estabelecer o processo de pensamento
 - Nada para definir os conceitos de livre-arbítrio, criatividade, ...
 - Dualismo (Descartes)
 - Natureza física VS mente
 - Livre-arbítrio

- Há uma parte da mente que não poderia ser explicada pelas leis da física
- Os animais não possuem esta qualidade de dualismo
- Materialismo
 - O mundo (cérebro e mente) funciona de acordo com leis físicas
 - Matéria → primeira e última substância de qualquer ser, coisa ou fenómeno do universo
 - A matéria pode compor tanto a pedra quanto os reinos animal e vegetal, e produzir efeitos surpreendentes como a luz, o som, a emoção e a consciência
- Empirismo (John Locke, David Hume, ...)
 - Fonte do conhecimento → Experiência, observação dos factos, generalização das regras
- Positivismo lógico (Circulo de Viena, Bertrand Russell, ...)
 - O conhecimento pode ser expresso em teorias lógicas
 - Todo significado de uma afirmação pode ser verificado ou falseado por meio de experimentação
- Matemática
 - Cronologia
 - Raciocínio dedutivo (Aristóteles)
 - 1760 - Probabilidade (Bayes)
 - 1840 - Formalização de operações lógicas (Boole)
 - 1880 - Lógica de primeira ordem, termo, predicado e quantificação (Frege)
 - 1900 - Formalização da matemática (Hilbert)
 - 1930 - Incompletude da aritmética (Gödel)
 - 1940 - Relação dos objetos da lógica com objetos do mundo (Tarski)
 - 1940 - Computabilidade (Turing e Church)
 - Computabilidade VS Complexidade
 - 1971 - Problemas NP-Completo (Cook)
 - Teoria de Jogos/Teoria da Decisão (Von Neumann)
- Psicologia
 - Cronologia
 - 1850 – Primeiro laboratório de psicologia experimental para o estudo da visão humana
 - Pesquisa baseada na introspecção dos sujeitos (subjetivismo)
 - 1900 – Behaviorismo
 - Observação da ação (reação) dos sujeitos
 - 1900 – Psicologia cognitiva
 - Metáfora computacional do cérebro
 - Elementos para uma teoria do comportamento humano
 - Crenças
 - Objetivos
 - Raciocínio
 - Processo de agir com base no conhecimento
 - 1) A ação é traduzida para uma representação interna
 - 2) A representação é manipulada por processos cognitivos para derivar novas representações internas
 - 3) Estas representações são re-traduzidas para uma ação

- Engenharia computacional
 - Hardware (aumento da velocidade de computação e capacidade de memória)
 - Software (linguagens, metodologias, interfaces, ...)
- Linguística
 - Cronologia
 - 1957 – Estruturas sintáticas (Chomsky)
 - Linguagem = estrutura das frases + conhecimento do mundo
 - Filosofia da linguagem → representação do conhecimento
 - Campo híbrido → processamento de linguagem natural ou linguística computacional
- Áreas de IA



- Exemplos de aplicação
 - Jogos (ex.: The Sims, FIFA Soccer, ...)
 - Controlo de tráfego aéreo
 - Busca de informação na web
 - Recomendação de produtos
 - Previsão
 - Detecção de intrusão e filtragem de spam
 - Sistema de controlo (ex.: sistema de controlo do carro)

- **AGENTES**

- Definições

- “Entidade que percebe seu ambiente através de sensores (ex.: câmaras, microfone, teclado, ...) e age sobre ele através de atuadores (ex. vídeo, impressora, braços, rodas, ftp, ...)” (Russel & Norvig, 2004)
 - “Sistema (hardware ou software) situado em um determinado ambiente, capaz de atuar de forma autónoma e racional dentro do ambiente para executar seus objectivos predeterminados” (Wooldridge, 2004)
 - “Programa de computador que funciona em background, e desenvolve tarefas autónomas conforme delegadas pelo utilizador” (River, 1996)
 - “Programas que desenvolvem diálogos, negociam e coordenam transferência de informações” (Michael Coen, 1996)
 - “Entidades que apresentam conceitos de habilidade para execução autónoma e habilidade para executar raciocínio orientado ao domínio” (Virdhagriswaran, 1995)

- Exemplos de agentes

- Agente humano
 - Sensores: Olho, ouvidos, pele, gosto, etc.
 - Actuadores: Mãos, pés, pernas, boca, etc.
 - Impulsionados pelos músculos.
 - Agente robot
 - Sensores: Câmara, infravermelhos, pára-choques, etc.
 - Actuadores: pinças, rodas, luzes, etc.
 - Usualmente impulsionados por motores.
 - Agente software
 - Sensores: Inputs
 - Actuadores: Outputs
 - Agente racional
 - Segue o princípio da racionalidade
 - Dada uma sequência perceptiva, o agente escolhe, segundo os seus conhecimentos, as acções que satisfazem melhor o seu objetivo
 - Racional != onisciente
 - Racional != clarividente
 - Racional != necessariamente bem sucedido
 - Limitações
 - Sensores
 - Actuadores
 - Raciocinador (conhecimento, tempo, etc.)
 - Agir para obter mais dados perceptivos é racional

- Medida de desempenho

- Critério que define o grau de sucesso de um agente na realização de uma dada tarefa
 - Dicotomias
 - Objetiva / Subjetiva
 - Quantitativa / Qualitativa
 - Momento de avaliação (antes ou depois)
 - Melhor / Pior / Média

- Propriedades dos agentes

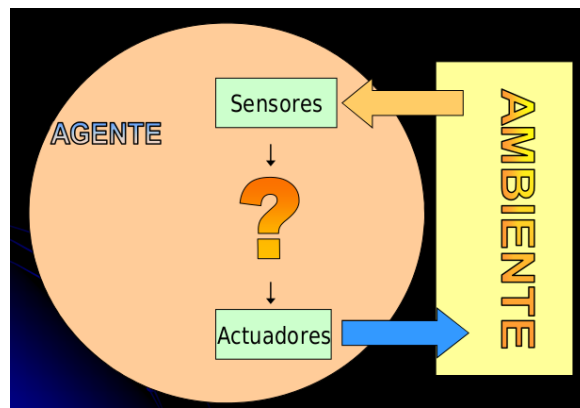
- Autonomia

- Operam sem a intervenção de humanos ou outros
- Possuem algum tipo de controlo sobre as suas acções e estado interno
- Pró-atividade
 - Não se limitam a agir em resposta ao seu ambiente
 - Tomar a iniciativa e exhibir comportamento direccionado por objectivos
- Reatividade
 - Têm a percepção do seu ambiente
 - Respondem rapidamente às alterações que nele ocorrem
- Habilidade Social
 - São capazes de interagir com outros agentes (e possivelmente com humanos) através de uma dada linguagem de comunicação de agentes
- Mobilidade
 - Capacidade de um agente se movimentar de um local para outro
- Conhecimento e Crença
 - Conhecimento → Colecção de informação dinâmica e capacidade de raciocínio sobre essa informação
 - Crença → Noção actual que o agente possui sobre um determinado facto
 - São geralmente dinâmicas (podem alterar o seu valor de verdade com o tempo)
- Intenções e Obrigações
 - Intenções → objectivos de longo prazo do agente
 - Obrigações → compromissos que o agente assumiu anteriormente
- Inteligência
 - O estado de um agente é formalizado por conhecimento (i.e. crenças, objectivos, planos e assunções) e ele interage com outros agentes utilizando uma linguagem simbólica
 - Capacidade de raciocínio abstracto e de resolução de novos problemas e adaptação a novas situações
- Continuidade temporal
 - Processo que é executado continuamente ao longo do tempo (persistente)
- Caráter
 - O agente possui uma personalidade credível e eventualmente possui também um “estado emocional”
- Aprendizagem
 - Adquire novo conhecimento
 - Altera o seu comportamento baseado na sua experiência prévia
- Estrutura dos agentes
 - Programa → função que relaciona as percepções (entradas) do agente para acções (saídas)
 - Arquitetura → suporte dado a execução dos programas de agente
 - A aquisição de percepções (sensores)
 - Execução de acções no ambiente (actuadores)
- Como definir agentes
 - Percepções
 - Ações
 - Objectivos
 - Ambiente
 - Tipos de ambiente

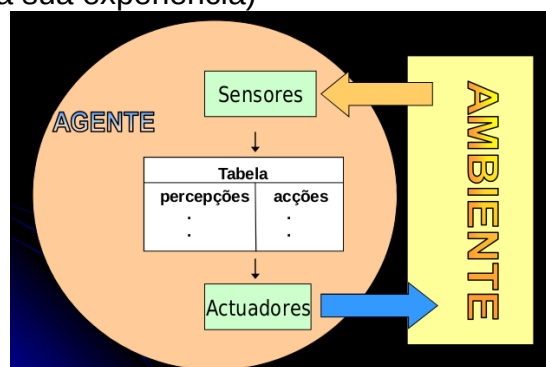
- Físico (ex.: robots)
- Software (ex.: softbot)
- Propriedades
 - acessível VS inacessível
 - acessível → os sensores do agente conseguem perceber o estado completo do ambiente
 - Exemplo → poker VS damas
 - estático VS dinâmico
 - estático → o ambiente não muda enquanto o agente está escolhendo a acção a realizar
 - semi-estático → o ambiente não muda enquanto o agente delibera, mas o "score" do agente muda
 - Exemplo → Civilization II VS Age of Empires
 - determinista VS não-determinista
 - determinista → o próximo estado do ambiente pode ser completamente determinado pelo estado actual e pelas acções seleccionadas pelo agente
 - Exemplo → backgammon VS damas
 - discreto VS contínuo
 - contínuo → as percepções e acções mudam num espectro contínuo de valores
 - Exemplo → ABS vs blackjack
 - episódico VS não-episódico
 - episódico → a experiência do agente é dividida em episódios (evento em o agente percebe e então age)
 - Exemplo → Roleta VS Xadrez
 - agente único VS agentes múltiplos
 - Exemplo → Palavras-cruzadas VS Xadrez
 - tamanho (do ambiente)
 - número de percepções, acções, objectivos

Agente/Ambiente	acessível	determinista	episódico	estático	discreto
xadrez sem relógio	Sim	Sim	Não	Sim	Sim
xadrez com relógio	Sim	Sim	Não	Semi	sim
gamão	sim	não	não	sim	sim
motorista de táxi	Não	Não	Não	Não	Não
médico	Não	Não	Não	Não	Não
tutor	Não	Não	Não	Não	Sim
Analizador de imagem	Sim	Sim	Sim	Semi	Não
Busca na web	Não	Não	Sim	Não	Sim
Filtrador de mail	Sim	Não	Sim	Não	Sim
Músico	Sim	Não	Não	Não	Não

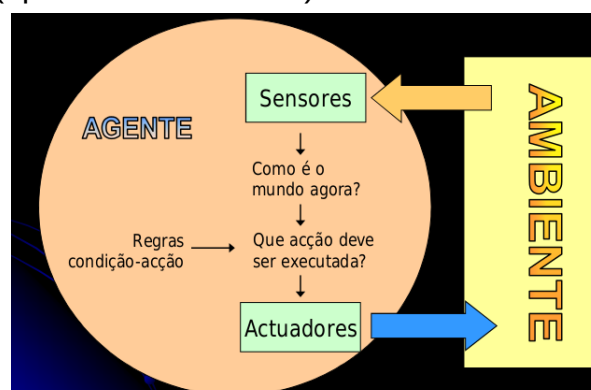
- Arquitetura dos agentes



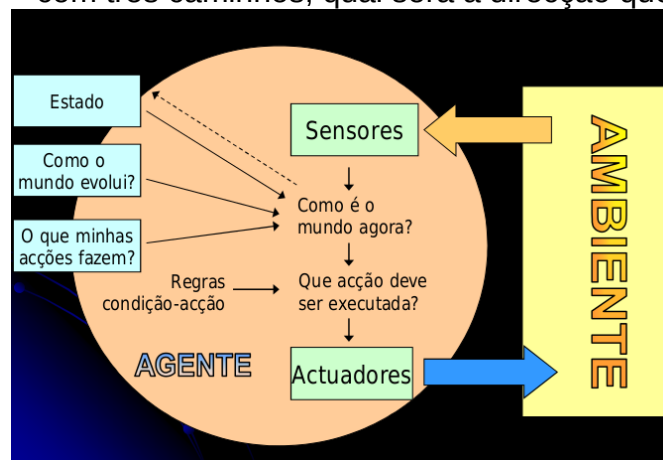
- Tipos de agentes
(- autonomia, - complexidade)
 - Agente tabela
 - Dada uma percepção procura a resposta (ação)
 - Ambiente acessível, determinista, episódico, estático, discreto e minúsculo
 - Problemas
 - A tabela pode ser muito grande
 - As acções são derivadas só do seu conhecimento interno (e não da sua experiência)



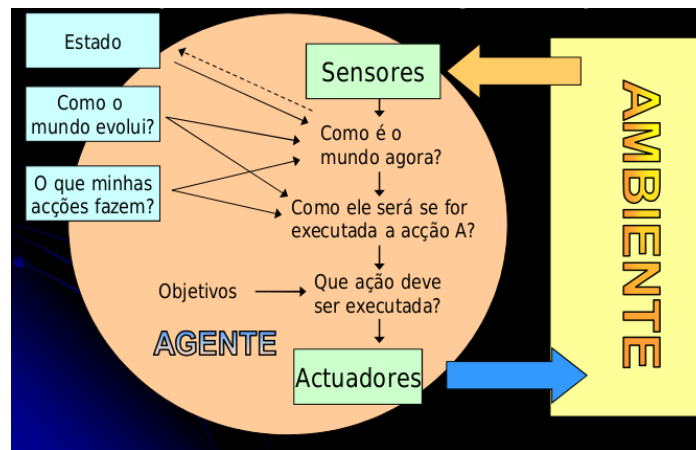
- Agente reativo
 - Sumarização de entradas/saídas comuns (uso de regras de condição)
 - Ambiente acessível, episódico, pequeno
 - Problemas
 - A decisão do agente só depende da percepção atual
 - Não pode armazenar uma sequência perceptiva, e tem pouca autonomia
 - Funciona apenas se o ambiente for completamente observável (aplicabilidade restrita)



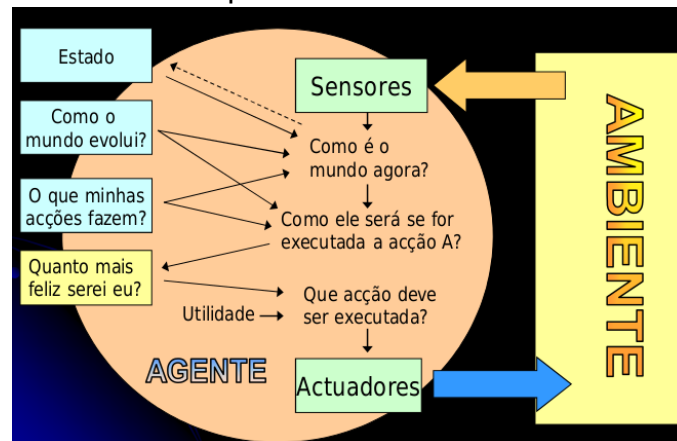
- Agente reactivo com estado interno (autómático)
 - O estado atual é dado em função do estado anterior/interno (histórico) e do que foi percebido no ambiente
 - Ambiente determinista e pequeno
 - Tipos de conhecimento necessários para atualizar a memória do agente
 - Como o ambiente evolui independentemente do agente (ex.: um carro que está ultrapassando em geral estará mais perto do que estava um instante antes)
 - Como as ações do próprio agente afetam o mundo (ex.: se o agente virar o volante à direita, o carro irá virar para a direita)
 - Problemas
 - Por vezes, mesmo com o estado interno, não se consegue tomar uma decisão (ex.: um motorista de táxi chega a um cruzamento com três caminhos, qual será a direcção que deve escolher?)



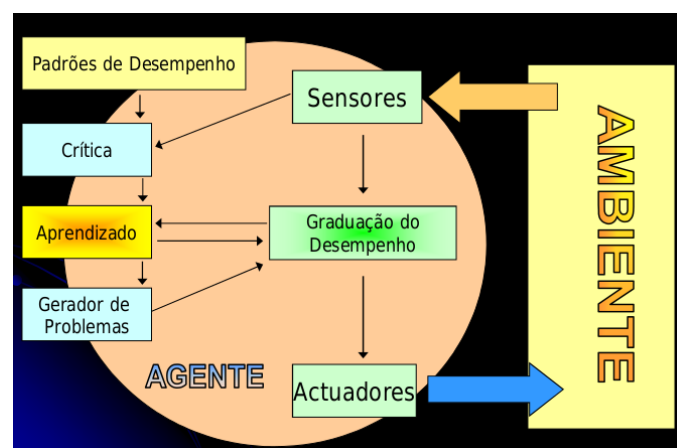
- Agente cognitivo (baseado em objectivos)
 - Escolhe acções que alcancem um dado objectivo
 - A seleção da ação pode ser direta (o resultado de uma única ação atinge o objetivo) ou mais complexa (o objetivo só é atingido após longas sequências de ações)
 - Para encontrar sequências de ações que alcancem o objetivo, são necessários algoritmos de busca (procura) e planeamento
 - Ambiente determinista
 - Problemas
 - algumas alternativas de planeamento de acções futuras podem ser mais rápidas, seguras ou baratas que outras



- Agente otimizador
 - Opta sempre pela ação que o levará a um estado mais desejável/com maior utilidade/que o deixará mais satisfeito
 - Ambiente sem restrição
 - Problemas
 - Não tem adaptabilidade



- Agente que aprende
 - Elementos
 - Elemento de aprendizado → responsável pela execução dos aperfeiçoamentos
 - Crítico → determina de que maneira o elemento de desempenho deve ser modificado
 - Elementos de desempenho → seleção de ações externas
 - Gerador de problemas → responsável por sugerir ações que levarão a experiências novas e informativas
 - Ambiente sem restrição

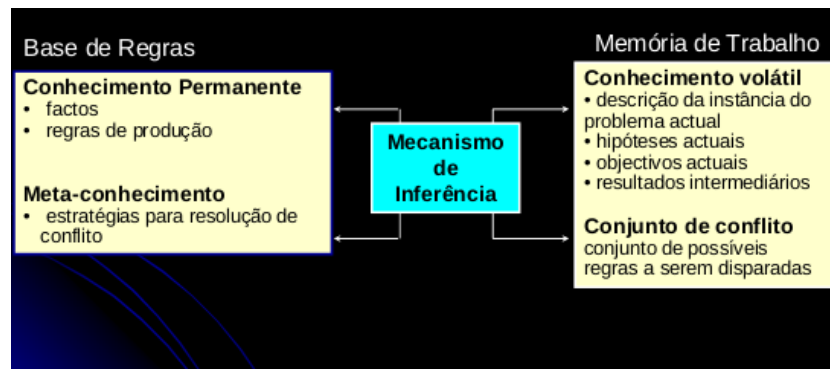


(+ autonomia, + complexidade)

- **PROCURA CEGA/PROCURA INFORMADA**

- Agente solucionador de problemas → agente inteligente que pode resolver problemas (considerando as diferentes sequências de ações possíveis) orientado a atingir metas particulares
 - Deve ter a representação do seu redor
 - Deve conhecer as ações que pode efetuar
 - Deve poder raciocinar sobre o efeito das suas ações
- Problema da representação → distintas formas de formular um problema e ineficiência para achar uma solução ao mesmo
- Modelo → interpretação de um dado domínio do problema segundo uma determinada estrutura de conceitos, representação de um problema em pequena escala
 - Ajudam a visualizar o problema
 - Permitem especificar a estrutura ou o comportamento de um problema
 - Permitem controlar e guiar o processo de resolução de um problema
- Características de uma boa representação
 - Clareza (deve ser evidente a relação entre o modelo e o problema real)
 - Exatidão (o modelo deve ser fiel à realidade)
 - Completude (o modelo deve apresentar todos os aspetos relevantes)
 - Eficiência
 - Conciso (o irrelevante e os detalhes devem ser omitidos)
 - Utilidade (o modelo sugere um bom modelo para a resolução do problema)
- Elementos de um problema em IA
 - Espaço de estados possíveis → Conjunto de todos os estados alcançáveis a partir do estado inicial por qualquer sequência de ações
 - Conjunto de ações (ou operadores) que permitem transitar de estado
 - Tipos de objetivos
 - Propriedade abstrata (ex.: condição de xeque-mate no Xadrez)
 - Conjunto de estados finais do mundo (ex.: estar no Porto Moniz)
 - Solução → caminho (sequência de ações ou operadores) que leva do estado inicial a um estado final (objetivo)
 - Custo do caminho → qualidade da solução
 - Procura → processo que gera/analisa sequências de ações para atingir um objetivo
 - Execução → executar a solução completa encontrada (procura cega, procura informada, estratégias com adversários)
 - Regras de produção → pares de condição-ação
 - Sistemas de produção → sistemas baseados em regras de produção
 - Elementos
 - Base de regras (BR) → conjunto de regras de produção
 - Memória de trabalho
 - Base de factos derivados durante a “vida” do agente
 - Percepções do agente e factos gerados a partir da BR pelo mecanismo de inferência
 - Mecanismo (máquina) de inferência
 - Determina o método de raciocínio utilizado (progressivo ou regressivo)
 - Utiliza estratégias de procura com compromisso (unificação)
 - Resolve conflitos e executa ações

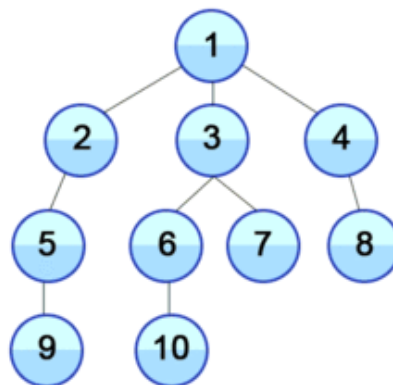
- Vantagens
 - Regras são de fácil compreensão
 - Inferência e explicações são facilmente derivadas
 - Manutenção mais simples, devido à modularidade
 - Maior eficiência do que os sistemas de programação em lógica (embora menos expressivos)
- Desvantagens
 - Conhecimento complexo requer muitas regras
 - Dificulta a utilização e manutenção do sistema
 - Não são robustos (tratamento de incerteza)
 - Não aprendem



- Aplicações na vida real
 - Cálculo de rotas
 - Rotas em redes de computadores
 - Sistemas de planeamento de viagens
 - Planeamento de rotas de aviões
 - Caixeiro viajante
 - Alocação (*scheduling*)
 - Salas de aula
 - Máquinas industriais (*job shop*)
 - Navegações de robots
 - Generalização do problema da navegação
 - Os robots movem-se em espaços contínuos, com um conjunto (infinito) de possíveis ações e estados
 - Montagem de objetos complexos por robots
 - Ordenar a montagem das diversas partes do objeto
- Procura em Espaço de Estados
 - Espaço de estados: árvore onde os estados são nós e as operações são arcos
 - Fronteira do espaço de estados: nós (estados) a serem expandidos no momento
 - Geração e Teste
 - 1) Começar por um nó
 - caso não haja, o algoritmo termina com falha
 - caso seja um nó final, o algoritmo termina com sucesso
 - 2) É gerado um novo conjunto de nós (ao expandir o nó)
 - 3) Volta para o passo 1)
 - Implementação genérica

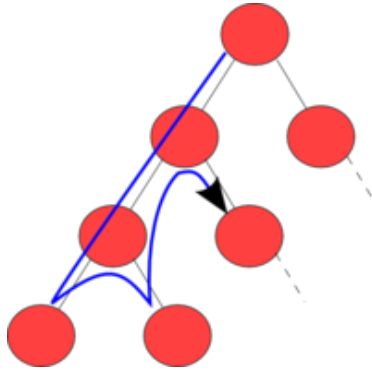
```
//Função-Inserir: controla a ordem de inserção de nós na fronteira do espaço de estados
procuraGenerica(problema)
{
    fronteira = faz-fila(faz-nó(problema[Estado-Inicial]));
    while(1)
    {
        if(!fronteira) return FALHA;
        nó = remove-primeiro(fronteira);
        if(forEstadoFinal(nó,problema)) return nó;
        fronteira = função-insere(fronteira,problema[operadores]);
    }
}
```

- Métodos de procura
 - Procura exaustiva-cega
 - São ineficientes na maioria dos casos
 - Não sabe qual o melhor nó da fronteira (melhor nó \Leftrightarrow menor custo de caminho desse nó até um nó final/objetivo)
 - Estratégia de procura: ordem da expansão dos nós
 - Direções de procura
 - Do estado inicial para o estado final
 - Do estado final para o estado inicial
 - Bidirecional
 - Algoritmos
 - Procura em largura (expande todos os nós da esquerda para a direita em cada nível)

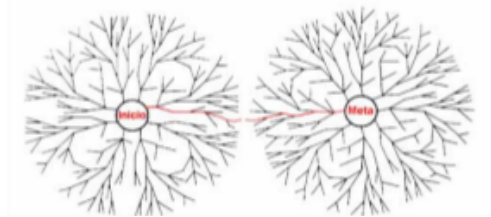


- Características
 - Completo
 - Ótimo
- Custo uniforme (igual à procura em largura, só que expandindo os nós por ordem ascendente do custo até esse nó)
 - Características
 - Completo
 - Ótimo (a solução que encontrar vai ser sempre a de menor custo)

- Procura em profundidade (expande todos os nós até o nível mais profundo (começando sempre pela sua esquerda (e dos sucessores)))



- Características
 - Não é completo
 - Não é ótimo
 - Menor custo de memória (não precisa de examinar a árvore por completo)
 - Maior custo de tempo
- Pode ser mais rápida do que procura em largura quando existem várias soluções
- Deve ser evitada quando as árvores geradas são muito profundas ou geram caminhos infinitos (senão, o algoritmo “desce” indefinidamente)
- Profundidade limitada (igual à procura em profundidade, só que o nível máximo de procura é fixo)
- Aprofundamento progressivo (várias iterações da profundidade limitada para valores crescentes do limite máximo)
- Bidirecional (procura em duas direções, que termina quando os dois processos geram um mesmo estado intermediário)

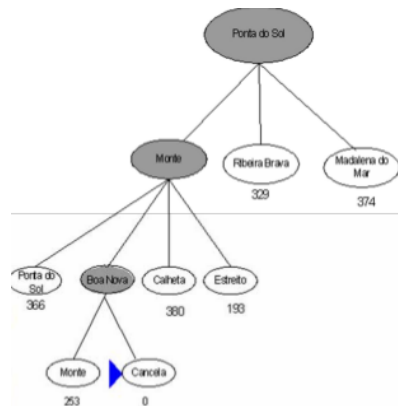


- Procura heurística-informada
 - Utilizam conhecimento específico do problema na escolha do próximo nó a ser expandido
 - Permite a libertação do uso das análises combinatórias (cortando certos caminhos)
 - Estima qual o melhor nó da fronteira a ser expandido com base em métricas (ex.: distância, custo, ...) - através de funções heurísticas(conhecimento)
 - As funções heurísticas são específicas para cada problema
 - É preciso ter um equilíbrio entre a eficiência da procura e o custo computacional
 - Notação
 - $g(n)$ → custo desde o nó inicial até n
 - $h(n)$ → custo estimado desde n até uma solução (resultado da

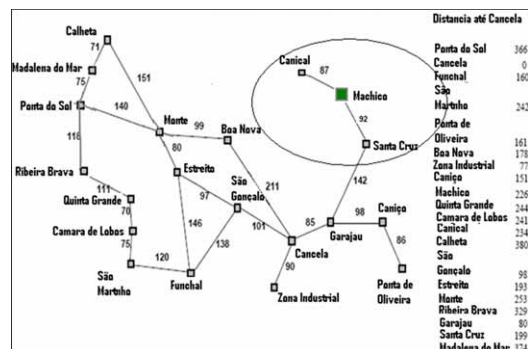
função heurística)

- $h^*(n) \rightarrow$ custo real desde n até uma solução
- $f(n) = g(n) + h(n) \rightarrow$ custo estimado desde o nó inicial até n
- Características
 - Admissibilidade (nunca sobreestima o custo real da solução – a estimativa é sempre menor ou igual ao custo)
 - **Consistência (??)**
 - Dominância (obtem estimativas maiores que outras funções heurísticas)
 - É recomendado utilizar uma função que domine todas as outras funções possíveis
 - **Qualidade (?? o quanto um processo de procura está concentrado em atingir o seu objetivo ??)**
 - Análise de Performance
 - Para a otimização são usados algoritmos de procura local (algoritmos em que só é considerado o estado atual e não são “olhados” os estados sucessores)
 - Vantagens
 - Usam muito pouca memória
 - Conseguem encontrar boas soluções em espaços de estados infinitos
 - Exemplos de algoritmos locais
 - Trepa-colinas (ciclo até atingir um pico (um máximo))
Problemas: máximos locais, planaltos, encostas e picos
Soluções: reinício aleatório (*random restart*), limite de iterações
 - Anelamento Simulado (trepa-colinas, só que em vez de reiniciar automaticamente, retrocede (tal como o cozimento dos metais))
- Estratégias de definição de heurísticas
 - Relaxação das regras do problema (não abrir muito a árvore)
 - Diminuição da complexidade das condições que envolvem a procura
 - Excluir qualquer consideração que possa ser considerada irrelevante para a resolução do problema
 - Efeitos
 - Favorece a admissibilidade
 - Favorece a consistência
 - Uso de informação estatística (recolhidos através dos resultados obtidos numa amostra da população)
 - Efeitos
 - Elimina a admissibilidade
 - **Identificação dos atributos mais relevantes do problema**
 - **Uso de algoritmos indutivos de aprendizagem**
- Estratégia de procura: melhor escolha baseada na função heurística
- Direções de procura
 - Do estado inicial para o estado final
 - Do estado final para o estado inicial

- Bidirecional
- Algoritmos
 - Best-First Search
 - Abordagens
 - Pesquisa Gulosa (expande os nós cujo estado é previsto como o mais perto do nó final, com base na estimativa feita pela função heurística)
 - Exemplo (percurso Ponta do Sol-Cancela, sendo $h(n)$ a distância até Cancela)



- Características
 - Não é completa (nós repetidos podem originar loop – ex.: Caniçal → Machico → Caniçal)

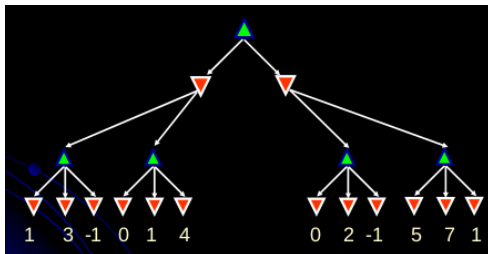


- Não é ótima (só escolhe o caminho que é mais curto à primeira vista)
- Maior custo de tempo (no pior caso, pode ter de expandir todos os nós)
- Maior custo de memória (mantém todos os nós em memória)
- Algoritmo A* (a avaliação de cada nó é $f(n)=g(n)+h(n)$, o qual queremos minimizar)
 - A condição $h(n) \leq h^*(n)$ tem que ser satisfeita
 - Características
 - Completo
 - Ótimo
 - Custo de tempo exponencial
 - Custo de espaço – guarda todos os nós expandidos na memória
- Algoritmo IDA* (igual ao algoritmo A*, mas com profundidade limitada (dado pelo valor de f))

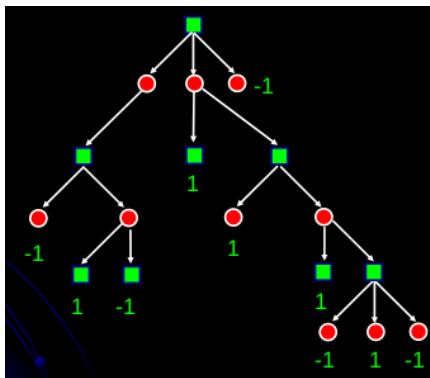
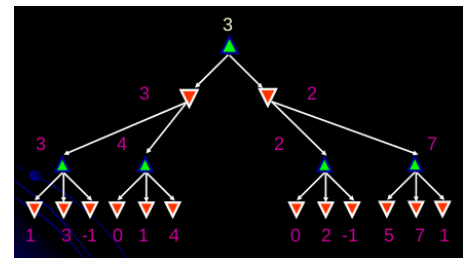
- ????
 - Algoritmo SMA* (igual ao algoritmo A* só que limitado em termos de memória)
 - ????
 - Críticas à busca heurística
 - Dificuldades em definir e usar a função de avaliação
 - Não consideram conhecimento genérico do mundo (o bom senso)
- Critérios de avaliação das estratégias de procura
 - Completude (“encontra sempre uma solução?”)
 - Custo de tempo (tempo demorado para encontrar a solução)
 - Custo de memória (memória necessária para a realização da procura)
 - Otimização/qualidade (“encontra a melhor solução entre várias?”)

- **BUSCA ADVERSÁRIA (JOGOS)**

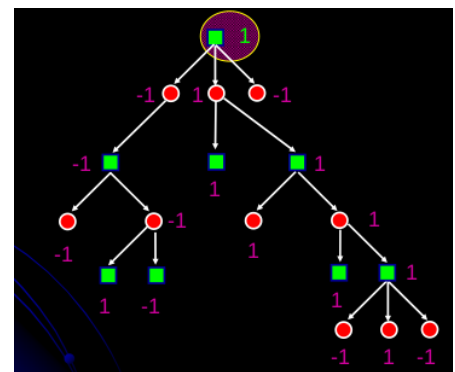
- Jogadores
 - Max (é o que começa o jogo)
 - Min
- Tipos de jogos
 - Informação
 - Perfeita: Xadrez, Damas
 - Imperfeita: Poker, Bisca
 - Sorte/Determinística
 - Determinístico: Xadrez, Damas
 - Sorte: Poker, Monopólio
- Características
 - Estado inicial (posição do tabuleiro e a indicação de quem deve jogar)
 - Conjunto de operadores (movimentos válidos à disposição dos jogadores)
 - Teste de terminação (verifica se o jogo está finalizado)
 - Função de utilidade (pontos do max)
 - ≥ 1 (max ganha)
 - 0 (empate)
 - ≤ -1 (min ganha)
- Algoritmo Minimax
 - Max realiza sempre a sua melhor jogada (maior valor de utilidade)
 - Min realiza sempre a pior jogada para o max (menor valor de utilidade)
 - É feita a procura cega em profundidade
 - Funcionamento



→

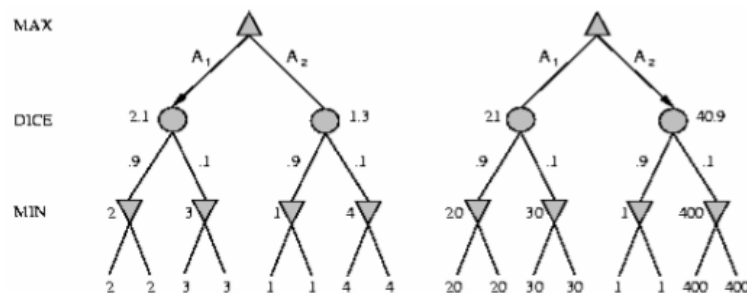


→



- Propriedades
 - Completo (se a árvore for finita)
 - Ótimo (contra um adversário ótimo)
 - Custo de tempo grande
- Poda Alpha-Beta
 - Alpha – melhor jogada para o Max (valor mais alto)

- Beta – pior jogada para o Max (valor mais baixo)
- Menor custo de tempo do que o Minimax
- Menor custo de memória do que o Minimax
- Funcionamento – igual ao Minimax, só que descartando certos nós desnecessários
 - Num nó Max, ao ser encontrado algum nó sucessor com valor positivo, os outros já são ignorados
 - Porque o Max ia escolher esse valor positivo (ou outro que surgisse, sendo indiferente para o resultado final)
 - Num nó Min, ao ser encontrado algum nó sucessor com valor negativo, os outros já são ignorados
 - Porque o Min ia escolher esse valor negativo (ou outro que surgisse, sendo indiferente para o resultado final)
- Problemas
 - Em casos extremos, pode continuar a gerar uma árvore inteira de pesquisa
 - Avaliação da utilidade usualmente não é exata
- Na maioria dos jogos determinísticos, o espaço de procura é muito grande e/ou existem restrições a nível de tempo
 - Portanto, o Minimax ou o Alpha-Beta deve ser alterado
 - A função de utilidade pode ser substituída por uma função de avaliação heurística - EVAL (que dá uma estimativa da utilidade)
 - Exemplos
 - Função de avaliação para o jogo de xadrez: soma dos valores das peças brancas em jogo menos a soma dos valores das peças negras em jogo
 - O teste terminal pode ser substituído por um teste de Cutoff (que decide quando aplicar a função EVAL)
 - Serve para parar a busca antes de chegar ao final (limite de custo)
 - Efeito horizonte – pode ser feito o corte na procura quando estaria uma jogada decisiva no nível seguinte
 - O corte na procura só deve ser feito num nó estável (com variância baixa/se encontra na zona de quietude)
- Jogos não determinísticos (jogos da sorte)
 - Não é possível construir uma árvore de jogo standard
 - É preciso ter em conta o factor sorte
 - Algoritmo ExpectiMinimax



- Constituição de uma árvore
 - Nós Max
 - Nós Min
 - Nós de Probabilidades
 - Cada nó sucessor tem associado uma probabilidade de ser

- pesquisado
- São calculados valores esperados baseados em probabilidades em vez de valores exatos
 - Valor esperado=média[(probabilidade de ser pesquisado)*utilidade]
- Características
 - Maior custo de tempo
 - Impossível aplicar os cortes Alfa-Beta aos nós (como não existe uma sequência exata)
- ?? Algoritmo *-Minimax (igual ao ExpectiMinimax, só que possibilita os cortes nos nós Max, nós Min e nós de probabilidades) ??

- **REPRESENTAÇÃO DO CONHECIMENTO E INFERÊNCIA**

- Bom senso
 - Componentes
 - Representação (cenário que possamos manipular, descrições que correspondam com o mundo ou com algum estado particular do mesmo)
 - Raciocínio (inferências a partir do cenário)
 - Abstração baseada em posturas
 - +concreto → +certo
 - +abstrato → +capacidade computacional
 - Posturas
 - Postura física (ao nível das ciências físicas e químicas - p.ex. massa, energia, velocidade, composição química, temperatura) – ex.: quando predizemos onde vai terminar um objeto que foi lançado
 - Postura de desenho (ao nível da engenharia e da biologia – p.ex. o propósito, a função, o desenho) – ex.: “um pássaro voa se mexe as asas, baseando-se em que as asas foram feitas para voar”
 - Postura intencional (crenças, desejos, metas, intenção) – ex.: “um pássaro voará porque um gato aproxima-se”
 - Conhecimento
 - Tipos de conhecimento
 - Conhecimento implícito VS conhecimento explícito
 - Conhecimento implícito – frases simbólicas que podem ser inferidas a partir do conhecimento explícito
 - Conhecimento explícito – frases simbólicas explicitamente armazenadas na base de conhecimento
 - Conhecimento soft VS conhecimento hard
 - Conhecimento comum VS conhecimento distribuído
 - Conhecimento interno VS conhecimento externo
 - Conhecimento canónico VS conhecimento não canónico
 - Conhecimento intencional VS conhecimento extensional
 - Conhecimento intencional – geral ao domínio da aplicação e classe de problema (restrições de integridade, regras de funcionamento do domínio, ...)
 - Conhecimento extensional – particular da instância do problema a resolver (proposições sobre instâncias de conceitos)
 - Conhecimento diagnóstico VS conhecimento casual
 - Conhecimento casual – prevê resultados de ações e eventos (ex.: olhando para o norte e virando à direita, passo a olhar para leste)
 - Conhecimento diagnóstico – forma hipóteses sobre causas de efeitos observados (ex.: como cheira a queimado, não devo ter desligado o fogão)
 - Conhecimento terminológico VS conhecimento dedutivo
 - Conhecimento terminológico – classificação de entidades do domínio e seus relacionamentos estáticos (ex.: Se João é filho do Pedro e Pedro é filho do Carlos, João é neto do Carlos)
 - Conhecimento dedutivo – regras de funcionamento e restrições de integridade do domínio (ex.: silogismos)
 - Conhecimento certo/preciso VS conhecimento incerto/vago

- Conhecimento certo/preciso – epistemologicamente booleano (ex.: Pedro é português)
 - Conhecimento incerto/vago – epistemologicamente probabilista (ex.: Se jogar um dado, vai sair um 6 com 1/6 de probabilidade)
- Conhecimento declarativo – descreve o que é conhecido acerca de um determinado problema (declarações que são assumidas como verdadeiras ou falsas e que descrevem um objeto ou conceito)
- Conhecimento procedimental – descreve como um problema é resolvido ou como agir perante uma dada situação (ex.: regras, estratégias, procedimentos)
- ?? Meta-conhecimento ??
- (...)
- Representação: relação entre dois domínios, onde um deles procura tomar o lugar do outro - ex.: o algarismo “4” (domínio simbólico) representa o número 4 (domínio abstrato dos números)
 - Usualmente, o domínio que representa é um domínio mais concreto (ou imediato ou acessível) que o domínio que é representado
- Um sistema inteligente utiliza estruturas que:
 - Podem ser interpretadas como proposições que representam o conhecimento do sistema
 - Determinam o comportamento do sistema
- (...)
- Componentes de um sistema de Conhecimento e Raciocínio
 - Linguagem formal (símbolos legais, especificando-se através de uma gramática)
 - Semântica (significado das sentenças da linguagem)
 - Teoria de raciocínio/prova (especifica como obter uma resposta da base de conhecimento)
- (...)
- Tipos de raciocínio não dedutivo
 - Raciocínio indutivo (raciocinar a partir de premissas particulares na busca de uma lei geral – p.ex. “todos os dias o sol aparece; amanhã também aparecerá”)
 - Raciocínio abdutivo (raciocinar através de diagnóstico ou da melhor explicação – p.ex. “sabemos que o carro não liga; pode ser que acabou a gasolina”)

- **SISTEMA MULTI-AGENTES**

- Razões para serem usados sistemas multi-agentes
 - Mais favorável para problemas complexos e de natureza descentralizada
 - Agentes individuais são limitados devido ao seu limitado conhecimento, recursos computacionais e perspectivas
- Origens
 - Sistemas Blackboard (Hearsay, Planner) – espaço de pesquisa partilhado onde se encontram os resultados obtidos (pelas fontes de conhecimento)
 - Modelo ACTOR – não existe memória comum e é centralizado no princípio da comunicação por mensagens
- Sistema no qual agentes autónomas interagem entre si de forma a resolver problemas que está para além das capacidades de um único indivíduo
- Vantagens
 - Maior rapidez na resolução de problemas (através do aproveitamento no paralelismo)
 - Diminuição da comunicação (transmite somente soluções parciais em alto nível para outros agentes ao invés de dados brutos para um lugar central)
 - Maior flexibilidade por ter agentes de diferentes habilidades (que são dinamicamente agrupados para resolver problemas)
 - Aumento da segurança (possibilidade de agentes poderem assumir responsabilidades de agentes que falham)
- Exemplos de problemas onde usar SMAs
 - Quando o domínio envolve distribuição característica dos dados, capacidades de resolução de problemas e responsabilidades
 - Quando há necessidade de manter a autonomia de subsecções, sem perda da estrutura organizacional
 - Quando há complexidade nas interações (negociação, partilha de informação e coordenação)
 - Quando há impossibilidade de descrição da solução do problema à priori (devido à possibilidade de perturbações em tempo real no ambiente e processos de negócio de natureza dinâmica)
- Elementos
 - Ambiente – E
 - Objetos – O
 - Conjunto de agentes – A
 - Conjunto de relações – R
 - Conjunto de operações – Op
 - Operadores que representam os resultados das operações em Op e as reações do ambiente a eles
- Características
 - Objetivo
 - Objetivo de longo prazo – consistir na dimensão computacional de uma possível ciência da interação
 - Pró-atividade (agir ativamente de acordo com o objetivo e não somente reativamente ao ambiente)
 - Habilidade social (capacidade de interagir com outros agentes (e possivelmente humanos), de forma a tomar a melhor decisão de ação)
- Áreas de aplicação
 - Entretenimento (jogos, teatro/cinema interativo)

- Aplicações industriais (ex.: ARCHON (controlo de processos))
- Controlo de tráfego aéreo (ex.: OASIS)
- Gestores de informação (ex.: Maxims, Newt)
- Comércio eletrónico (ex.: Kabash (sistema que cria agentes de compra e venda, onde as transações são realizadas através da interação entre esses agentes))
- Aplicações médicas (ex.: GUARDIAN (sistema de monitorização de pacientes em cuidado intensivo pós-operatório))
- Perspetivas
 - Perspetiva do agente
 - Categorias do agente
 - Estrutura e manutenção do conhecimento
 - Estruturas de conhecimento
 - Factos/crenças
 - Metas/Intenções
 - Preferências
 - Motivações
 - Desejos
 - ...
 - Capacidade de raciocínio (raciocinar sobre obrigações, permissões, interdições, crenças, desejos, intenções, conhecimento e comportamento de outros agentes, ...)
 - Capacidades de adaptação e aprendizagem (adaptação em relação ao ambiente e ao comportamento de outros agentes)
 - Arquiteturas de agentes
 - Agentes reativos simples (usam apenas um conjunto de regras situação-ação)
 - Agentes memorizando o mundo (a decisão implica um conhecimento prévio do mundo)
 - Agentes geridos por objetivos (usa informação sobre os objetivos, implica pesquisa e planeamento)
 - Agentes baseados na utilidade (a utilidade é usada para decidir entre objetivos em conflito (quando há incerteza em ações))
 - Perspetiva do grupo
 - Organização
 - Coordenação
 - Cooperação
 - Negociação
 - Comportamento coerente
 - Planeamento
 - Comunicação
 - Interação