

Sistemas inteligentes

Inteligência artificial

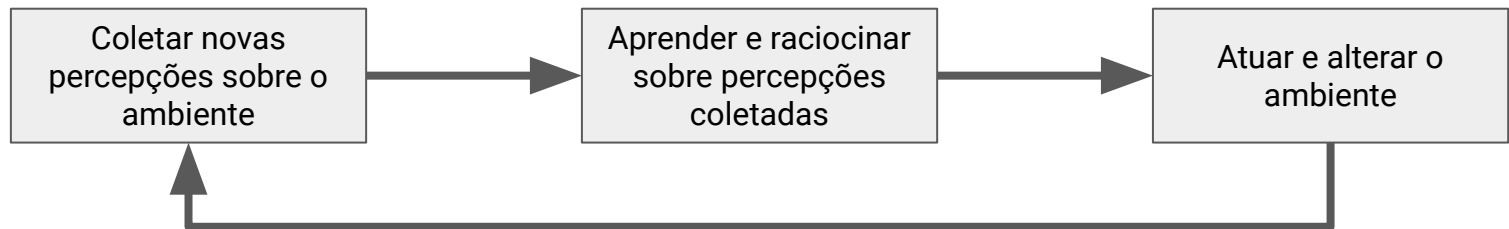
Prof. Allan Rodrigo Leite

Sistemas inteligentes

- Um sistema é considerado inteligente quando é capaz de executar tarefas que humanos fariam
 - Sistemas complexos não são necessariamente inteligentes
 - Um sistema de controle de um braço mecânico que repete uma sequência de movimentos não deveria ser considerado inteligente
- Características de um sistema inteligente
 - Aprendizado e auto-organização
 - Adaptação em um ambiente ou situação desconhecida
 - Raciocínio e reação às percepções coletadas sobre o ambiente

Sistemas inteligentes

- Características de um sistema inteligente (cont.)



- Relembrando as abordagens de raciocínio

- Simbólica
- Conexionista
- Evolutiva
- Estatística

Agentes baseados em conhecimento

- Abordagens baseadas em busca
 - São eficientes quando o problema pode ser formalizados por
 - Conjunto de estados e ações (espaço de estados)
 - Estado inicial
 - Conjunto de estados alvo
 - Porém, não são capazes de resolver problemas que exigem raciocínio a partir de algum conhecimento sobre o domínio do problema

Agentes baseados em conhecimento

- Abordagens baseadas em busca (cont.)
 - Nesta abordagem o conhecimento do domínio é fraco e limitado
 - Normalmente o conhecimento representado por heurísticas na busca
 - Diversos problemas necessitam prever um conhecimento explícito
 - Sistemas para diagnóstico médico
 - Controle de tráfego aéreo
 - Prova de teoremas
 - Sistemas envolvendo probabilidade

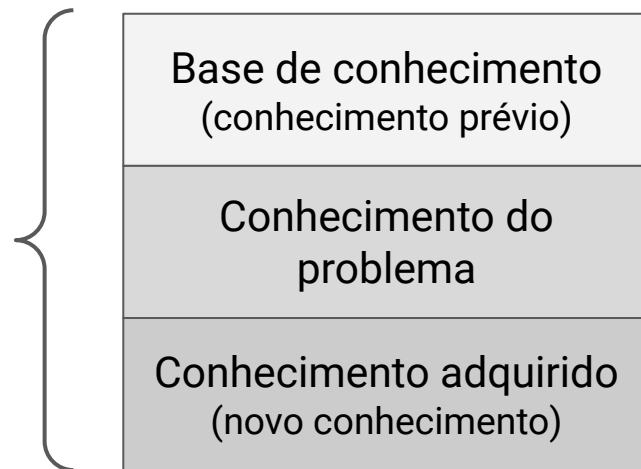
Agentes baseados em conhecimento

- Agentes baseados em conhecimento
 - Uma das linhas de pesquisa da abordagem simbólica
 - The Knowledge Principle (Lenat & Feigenbaum, 1991)
 - If a program is to perform a complex task well, it must know a great deal about the world in which it operates
 - Agentes baseados em conhecimento requerem
 - Identificar o conhecimento do domínio
 - Representá-lo em uma linguagem formal
 - Implementar um mecanismo de inferência para utilizá-lo

Agentes baseados em conhecimento

- Desafios envolvendo agentes baseados em conhecimento
 - Adquirir conhecimento sobre o domínio do problema
 - Representar adequadamente o conhecimento
 - Raciocinar corretamente e de forma eficiente a partir do conhecimento

Representação do conhecimento



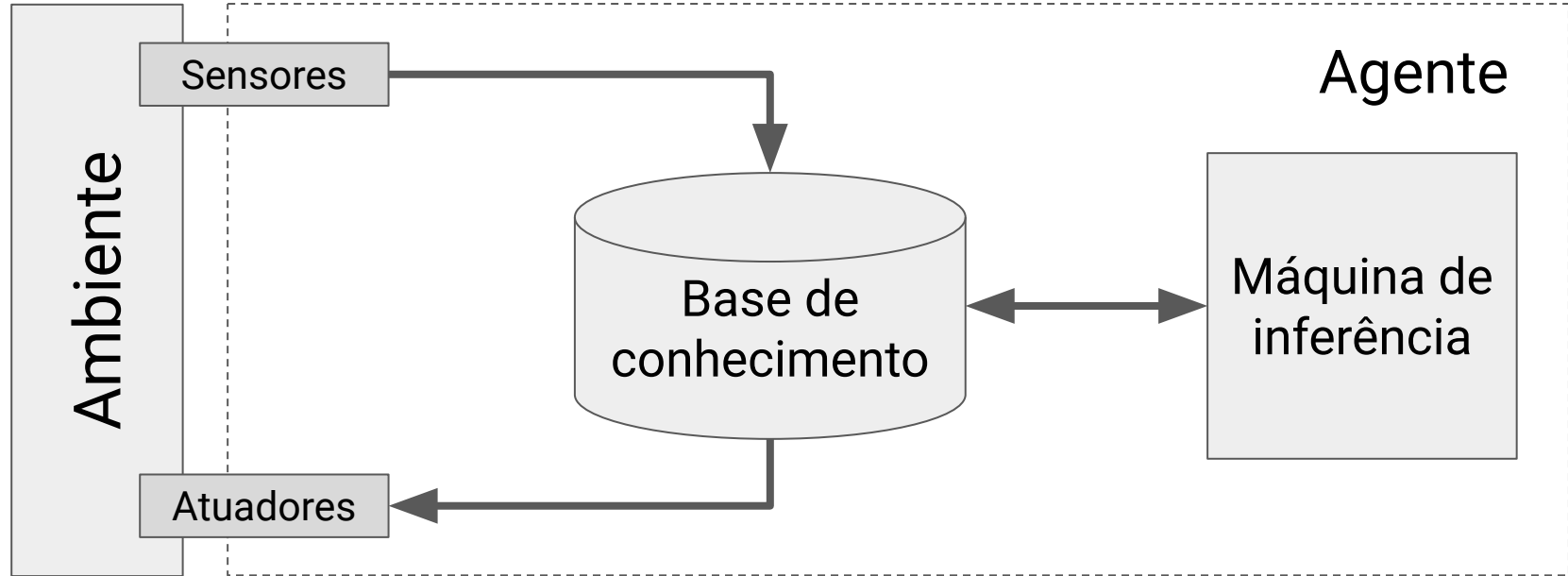
Agentes baseados em conhecimento

- Princípios
 - Conhecem seu mundo, isto é, o domínio do problema
 - Raciocinam sobre as possíveis ações
- Características
 - Conhecem o estado atual do mundo
 - Sabem como o mundo evolui
 - Conhecem estados desejáveis do mundo, ou seja, o estado alvo
 - Podem planejar e avaliar o resultado das ações
 - Capazes de planejar as ações
 - Podem utilizar metaconhecimento (aprendem como aprender)

Agentes baseados em conhecimento

- Composto minimamente por dois componentes básicos e separados
 - Base de conhecimento
 - Mecanismo de inferência
- Base de conhecimento
 - Representações de regras e fatos sobre o domínio do problema
 - Utiliza uma linguagem de representação do conhecimento
 - Compreensível pelo computador e preferencialmente de fácil manutenção
- Mecanismo ou máquina de inferência
 - Infere novos fatos ou hipóteses intermediárias ou temporárias
 - Utiliza a base de conhecimento para fundamentar as inferências

Agentes baseados em conhecimento



Categorias de raciocínio

- Dedução

- Fatos combinados com regras implicam em novos fatos (causa e efeito)
- É o único tipo de inferência que preserva a verdade (*truth-preserving*)
- Exemplo
 - Se há fogo (causa), há fumaça (efeito)
 - Aqui tem fogo, logo, aqui tem fumaça (novo fato)

- Abdução

- Inverso da dedução, isto é, do efeito para a causa
- Este tipo de inferência preserva a falsidade
- Exemplo
 - Se há fumaça, há fogo
 - Eu vi fumaça (efeito), logo aqui tem fogo (causa)

Categorias de raciocínio

- Indução
 - Parte dos fatos observados para gerar regras
 - Transforma o conhecimento em extensão em conhecimento em intenção
 - Relembrando
 - Conhecimento em intenção: definição de um conceito
 - Conhecimento em extensão: instância de um conceito
 - Exemplo
 - Maria observou a presença de fumaça e fogo em dias passados
 - José observou fumaça, então conclui-se que há fogo

Categorias de raciocínio

- Raciocínio analógico
 - Não oferece certeza, mas sim uma probabilidade
 - Combina fatos, similaridades e regras de adaptação
 - Resolve o problema a partir de fatos e da similaridade entre eles
 - Exemplo
 - Em algum outro caso de dengue, foi receitado aspirina e não deu certo
 - Logo, vou evitar receitar aspirina neste novo caso semelhante

Categorias de raciocínio

- Aplicabilidade das categorias de raciocínio
 - Dedução e abdução (via dedução)
 - Usadas em agentes baseados em conhecimento declarativo
 - Indução e analogia
 - Usadas em aprendizagem de máquina
 - Dedução (dois grandes grupos)
 - Usadas em problemas formulados em linguagem declarativa (lógica e afins)
 - Usadas em cenários incertos, probabilísticos ou nebulosos (*fuzzy*)

Sistemas de produção

- Sistema de representação procedimental
 - Regras de produção representam conhecimento por meio de um conjunto de regras do tipo **se** <condição> **então** <ação>
 - A ação corresponde a algum procedimento
 - Este procedimento leva a uma conclusão ou mudança no estado corrente
- Regras de produção
 - Contém um fragmento independente do conhecimento
 - Cujo conhecimento pode ser refinado com a adição de uma nova regra

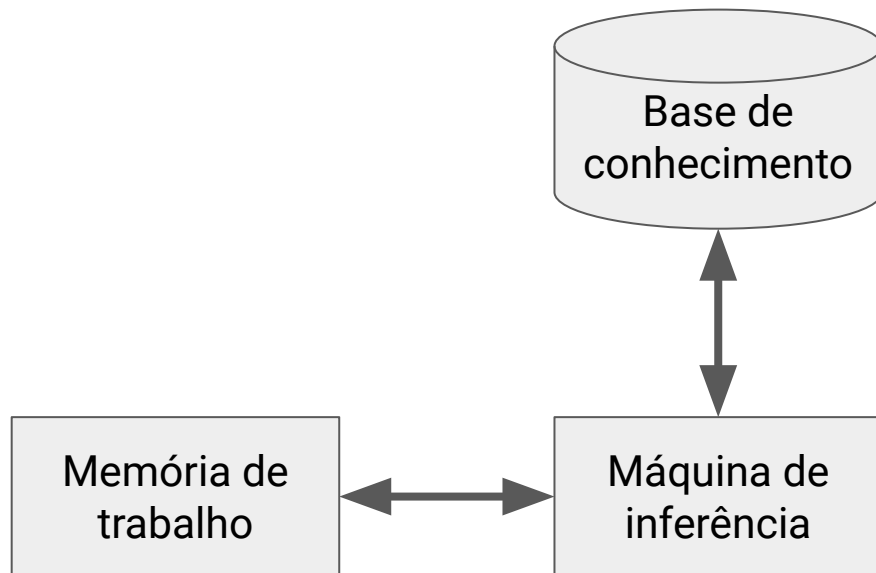
Sistemas de produção

- Regras de produção (cont.)
 - Condição da regra
 - Determina quando uma regra pode ser aplicada para a situação do problema
 - Também chamada de *left hand side* (LHS)
 - Ação da regra
 - Define o próximo passo na busca da solução para o problema
 - Também chamada de *right hand side* (RHS)

Sistemas de produção

- Um sistema de produção é formado por
 - Base de conhecimento composta por regras e fatos
 - Regras são declarações sobre classes e objetos se então
 - Fatos são declarações sobre objetos específicos
 - Memória de trabalho
 - Representa o estado do problema em um dado momento
 - Manipula dados transientes e de curta duração
 - Existem enquanto uma dada regra estiver sendo interpretada
 - Máquina de inferência
 - É acionada ao ser especificado um estado meta
 - Executa regras e determina quais são relevantes a partir de ciclos

Sistemas de produção



Sistemas de produção

- Um ciclo da máquina de estados pode ser dividido em três etapas
 - Seleção de regras (casamento)
 - Busca as regras que são satisfeitas pelo conteúdo da memória de trabalho
 - Resolução de conflitos
 - Usa estratégias para resolver conflitos no casamento das regras
 - As principais são raciocínio progressivo (regras) ou regressivo (metas)
 - Ação
 - Procedimento a ser realizado após o casamento e resolução de conflitos
 - Normalmente altera o estado da memória de trabalho

Sistemas de produção

- O motor de inferência pode usar duas abordagens para raciocínio
 - Raciocínio progressivo
 - Raciocínio regressivo
- Raciocínio progressivo
 - O encadeamento dos fatos e regras é para a frente
 - Também chamado de *data-driven inference* (dos dados à conclusão)
 - Características
 - As regras da base de conhecimento são usadas para gerar novos fatos
 - A geração dos novos fatos ocorre a partir de um conjunto inicial de dados
 - Os fatos gerados passam a fazer parte da base de conhecimento

Sistemas de produção

- Raciocínio regressivo
 - Raciocínio guiado da hipótese aos dados
 - Também conhecido como *goal-directed inference*
 - Características
 - Usa as regras da base de conhecimento para responder perguntas
 - Em outras palavras, tenta provar se uma asserção é verdadeira
 - Só processa as regras relevantes para a pergunta (asserção)

Sistemas de produção

- Resolução de conflitos
 - Regra #1: **se** <sinal> = verde **então** <ação> ← continue
 - Regra #2: **se** <sinal> = vermelho **então** <ação> ← pare
 - Regra #3: **se** <sinal> = vermelho **então** <ação> ← continue
 - Estratégias possíveis
 - Parar quando o objetivo for alcançado
 - Regra com maior prioridade
 - Regra mais específica
 - Regra mais recente

Sistemas de produção

- Ferramentas para desenvolvimento de sistemas de produção
 - CLIPS (*C Language Integration Production System*)
 - Ambiente para construção de sistemas especialistas baseados em regras
 - Projeto *open-source*
 - Jess (*Java Expert System Shell*)
 - Usa a mesma sintaxe do CLIPS
 - Projeto não é gratuito para uso comercial
 - Drools
 - Motor para regras de negócio baseadas em regras de produção
 - Projeto *open-source*

Exercícios

1. Implemente um sistema de produção para resolver o teste do Einstein. Há 5 casas de 5 diferentes cores e em cada casa mora uma pessoa de uma diferente nacionalidade. Os 5 proprietários bebem diferentes bebidas, fumam diferentes cigarros e têm um animal de estimação. Nenhum deles possui o mesmo animal nem fumam o mesmo cigarro ou bebem a mesma bebida. (continua)

Exercícios

1. O sistema deve responder perguntas como: "Qual é o animal de estimação do Inglês?". Considere as dicas abaixo para construção da base de conhecimento.
 - O Inglês vive na casa Vermelha.
 - O Sueco tem Cachorros como animais de estimação.
 - O Dinamarquês bebe Chá.
 - A casa Verde fica do lado esquerdo da casa Branca.
 - O homem que vive na casa Verde bebe Café.
 - O homem que fuma Pall Mall cria Pássaros.
 - O homem que vive na casa Amarela fuma Dunhill.
 - O homem que vive na casa do meio bebe Leite.
 - O Norueguês vive na primeira casa.
 - O homem que fuma Blends vive ao lado do que tem Gatos.
 - O homem que cria Cavalos vive ao lado do que fuma Dunhill.
 - O homem que fuma BlueMaster bebe Cerveja.
 - O Alemão fuma Prince.
 - O Norueguês vive ao lado da casa Azul.
 - O homem que fuma Blends é vizinho do que bebe Água.

Sistemas *fuzzy*

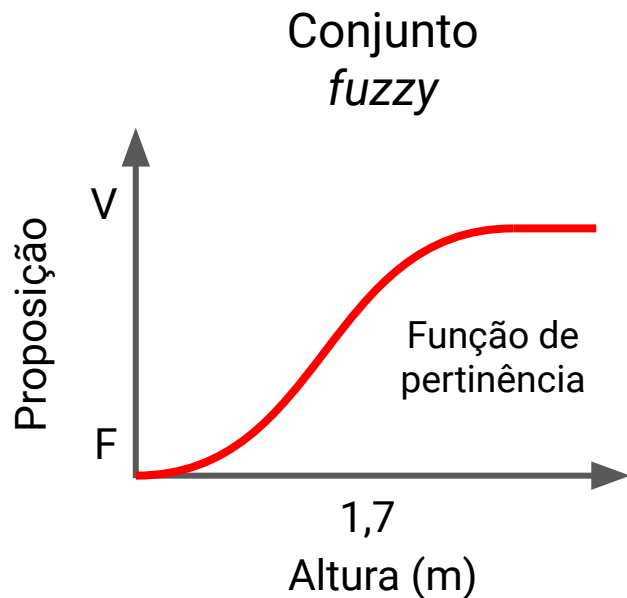
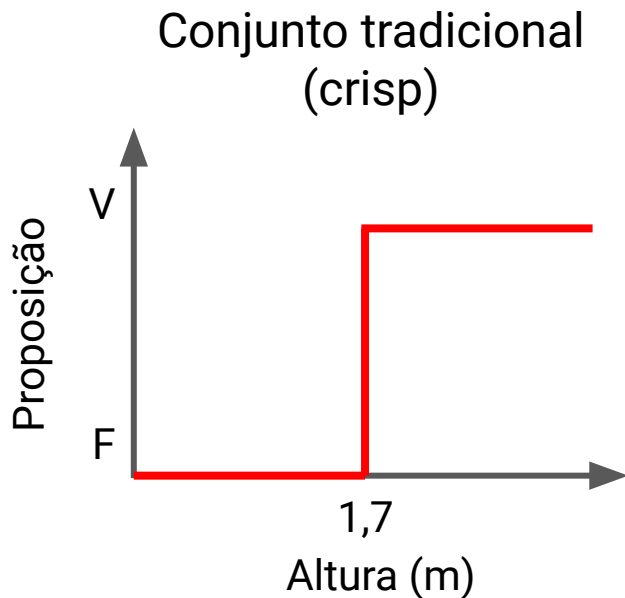
- São sistemas baseados na teoria de conjuntos *fuzzy* e lógica *fuzzy*
 - Também utilizam um conjunto de regras **se** <condição> **então** <ação>
 - Porém, o mecanismo de inferência permite um raciocínio aproximado
- Teoria de conjuntos *fuzzy*
 - Uma proposição lógica tem dois extremos: verdadeiro ou falso
 - Na lógica fuzzy, uma premissa varia em grau de verdade de 0 a 1
 - 0 que pode representar parcialmente verdadeira ou parcialmente falsa
 - Usa uma técnica baseada em graus de pertinência (verdade)
 - Os valores 0 e 1 ficam na extremidade
 - Prevê vários estados de verdade entre 0 e 1
 - A ideia por trás é que todas as informações admitem um grau

Sistemas *fuzzy*

- Teoria de conjuntos *fuzzy* (cont.)
 - Permite especificar o quão bem algo satisfaz uma proposição vaga
 - Exemplo: eu sou alto
 - Como interpretar o termo "alto"?
 - Esta proposição é verdadeira para alguém com 1,70 m?
 - Vantagens
 - Permite capturar melhor o que as pessoas pensam
 - Modela critérios mais subjetivos para tomada de decisão ou senso comum
 - Trabalha com uma variedade de informações vagas e incertas
 - Exemplo: maioria, mais ou menos, talvez, ...

Teoria de conjuntos *fuzzy*

- Relação entre conjuntos tradicionais e conjuntos *fuzzy*



Teoria de conjuntos *fuzzy*

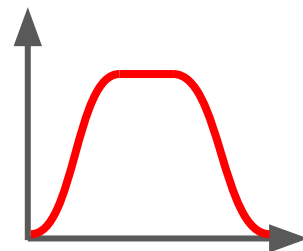
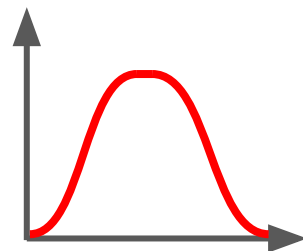
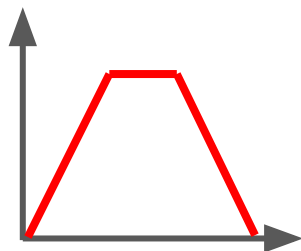
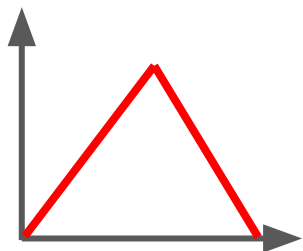
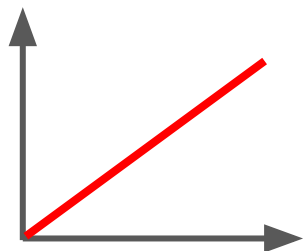
- Função de pertinência \mathcal{U}_A
 - Descreve um conjunto *fuzzy* definido no universo \mathcal{X}
 - A função mapeia os elementos de X para o intervalo $[0, 1]$
 - $\mathcal{U}_{A:X} \rightarrow [0, 1]$
 - Associa cada elemento y pertencente à \mathcal{X} a um número real entre $[0, 1]$
 - A associação representa o grau de pertinência do elemento y ao conjunto A
 - Ou seja, o quanto é possível para o elemento y pertencer ao conjunto A
 - Portanto, uma sentença pode ser
 - Parcialmente verdadeira
 - Parcialmente falsa

Teoria de conjuntos *fuzzy*

- Função de pertinência $\mathcal{U}_A(x)$
 - Indica o grau de compatibilidade entre x e o conceito expresso por A
 - $\mathcal{U}_A(x) = 1$: indica que x é completamente compatível com A
 - $\mathcal{U}_A(x) = 0$: indica que x é completamente incompatível com A
 - $0 < \mathcal{U}_A(x) < 1$: indica que x é parcialmente compatível com A
 - x é parcialmente compatível levando em consideração um grau $\mathcal{U}_A(x)$
- $A = \{(x, \mathcal{U}_A(x)) \mid x \in \mathcal{X}\}$
- Onde:
 - A é o conjunto *fuzzy*
 - \mathcal{U}_A é a função de pertinência
 - \mathcal{X} é o universo

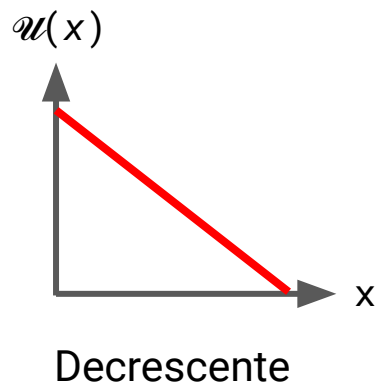
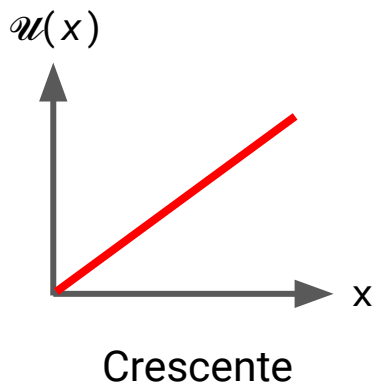
Teoria de conjuntos *fuzzy*

- Função de pertinência $\mu_A(x)$ (cont.)
 - Permite representação de universos discretos ou contínuos
 - Pode apresentar diferentes comportamentos (dinâmicas)
 - Linear
 - Triangular ou trapezoidal
 - Gaussiana
 - Sino generalizada
 - Quadrática



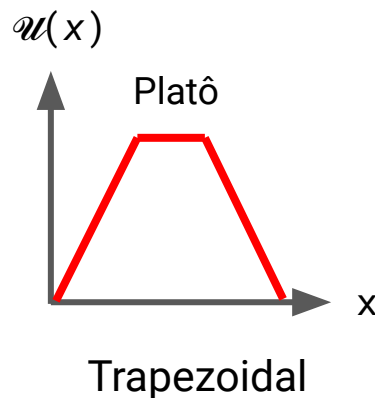
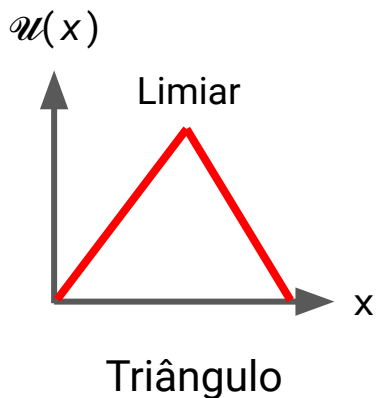
Teoria de conjuntos *fuzzy*

- Função de pertinência linear
 - Função com comportamento mais simples
 - Aproximação apresenta uma noção de crescente ou decrescente



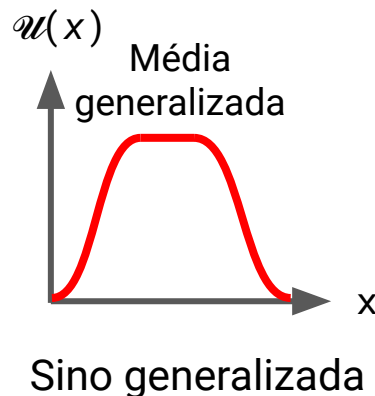
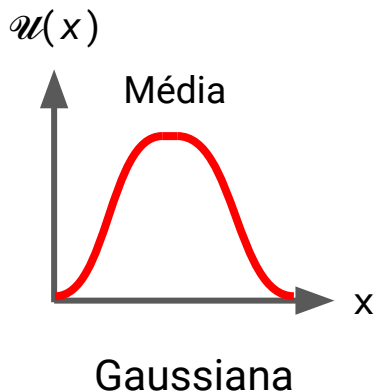
Teoria de conjuntos *fuzzy*

- Função de pertinência triangular e trapezoidal
 - Triangular
 - Apresenta mudança de comportamento em um dado limiar
 - Trapezoidal
 - Apresenta mudança de comportamento entre um platô
 - Adequada para uma aproximação que mapeia a expressão "em torno"



Teoria de conjuntos *fuzzy*

- Função de pertinência gaussiana e sino generalizada
 - Apresenta uma mudança suavizada de comportamento
 - Aproximação que mapeia uma distribuição normal
 - Tendência para um valor média

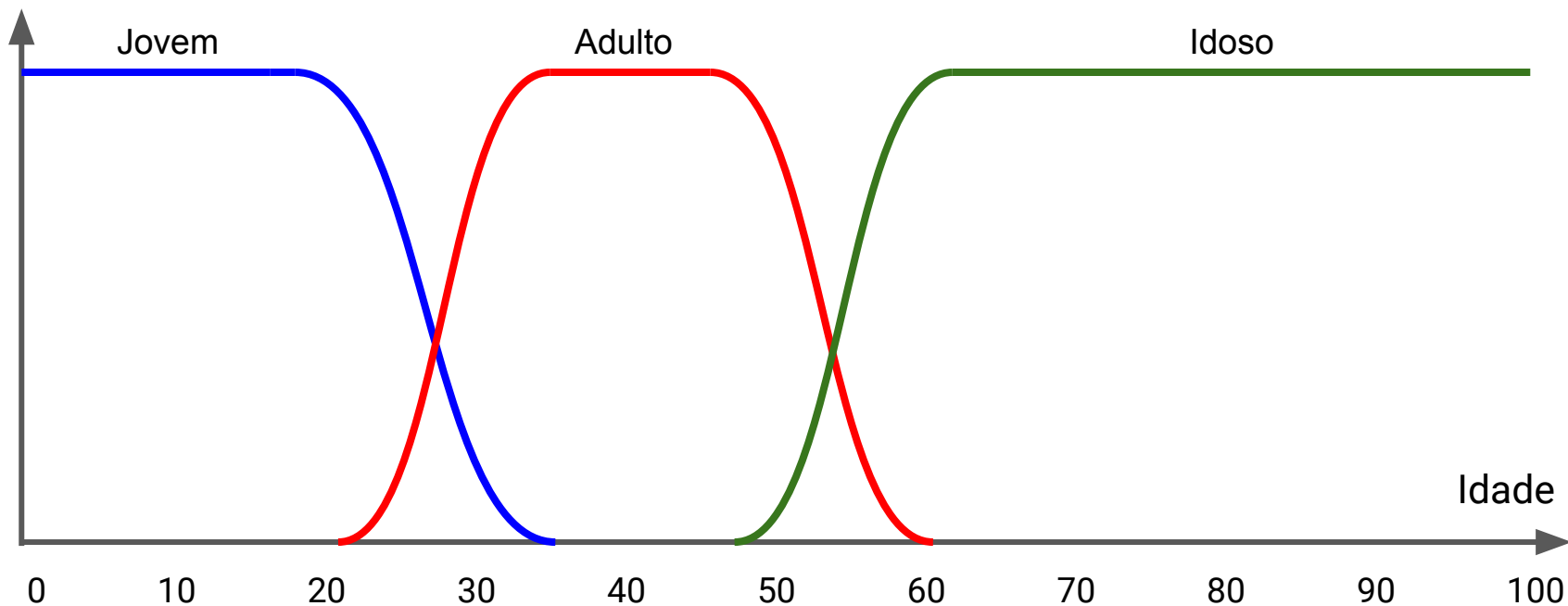


Teoria de conjuntos *fuzzy*

- Variável linguística
 - Representa uma partição do universo de discurso \mathcal{X}
 - Possui valores que não são números
 - São palavras, rótulos ou frases em linguagem natural
 - Exemplo: idade = jovem, idade = idoso, ...
 - Um valor linguístico é um conjunto *fuzzy*
 - Todos os valores linguísticos formam um conjunto de termos
 - $T(\text{idade}) = \{\text{jovem}, \text{adulto}, \text{velho}, \text{não jovem}, \pm \text{velho}, \dots\}$
 - Faz a modelagem *fuzzy* expressar a semântica usada por especialistas
 - Monitoramento de equipamentos em um sistema de controle de produção
 - **se** <ruído = um pouco baixo> **então** <consumo = eficiente>

Teoria de conjuntos *fuzzy*

- Variável linguística (cont.)



Teoria de conjuntos *fuzzy*

- Operações sobre conjuntos *fuzzy*
 - NÃO-fuzzy(x) = $1 - x$
 - Negação da sentença original
 - E-fuzzy(x, y) = $\min(x, y)$
 - Junção de duas sentenças
 - Forma uma conjunção de duas sentenças
 - OU-fuzzy(x, y) = $\max(x, y)$
 - Disjunção de duas sentenças

Teoria de conjuntos *fuzzy*

- Exemplo: representação de forma *fuzzy*
 - Alice com 1,65 m
 - Bob com 1,75 m
 - Carol com 2,0m
 - Denis com 1,45 m
- As proposições são "X é alto" com as seguintes pertinências
 - A = Alice é alta, $\mu(A) = 0,55$
 - B = Bob é alto, $\mu(B) = 0,75$
 - C = Carol é alta, $\mu(C) = 1,0$
 - D = Denis é alto, $\mu(D) = 0$

Teoria de conjuntos *fuzzy*

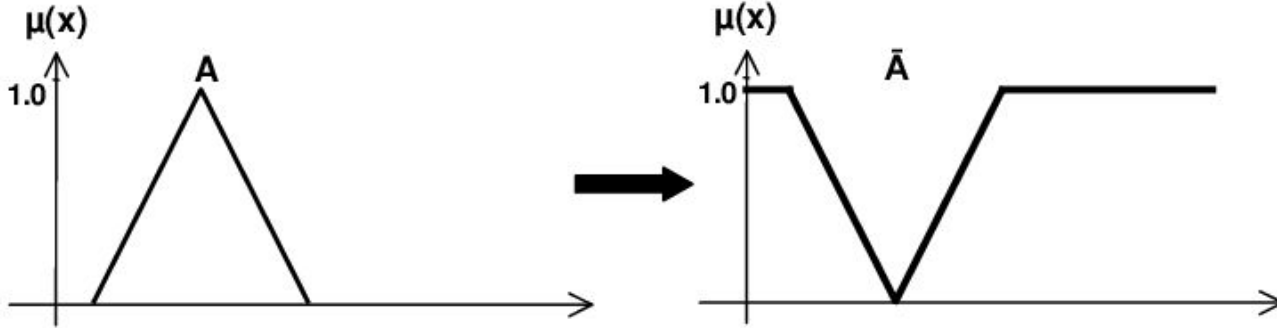
- Exemplo: representação de forma *fuzzy* (cont.)
 - Usando operadores fuzzy é possível escrever as sentenças
 - Carol não é alta
 - $\text{NÃO}(C)$
 - $\mathcal{U}(\text{NÃO}(C)) = 1,0 - \mathcal{U}(C) = 0,0$
 - Bob não é alto
 - $\text{NÃO}(B)$
 - $\mathcal{U}(\text{NÃO}(B)) = 1,0 - \mathcal{U}(B) = 0,25$
 - Denis é alto e Alice é alta
 - $\mathcal{U}(D)$ e $\mathcal{U}(A)$
 - $\mathcal{U}(D \text{ e } A) = \min(\mathcal{U}(D), \mathcal{U}(A)) = 0,0$

Lógica *fuzzy*

- A teoria de conjuntos *fuzzy* oferece suporte para a lógica *fuzzy*
 - Produz valores de saída sem a necessidade de entradas precisas
 - Os valores de saída são aproximações para lidar com incertezas
 - Também conhecido como raciocínio aproximado
- Lógica *fuzzy*
 - Conjunto de regras expressas através de implicações lógicas
 - Estas implicações são do tipo **se** <condição> **então** <ação>
 - Similar à implicação $\langle A \rightarrow B \rangle$ da lógica tradicional
 - Combina regras com os demais operadores dos conjuntos *fuzzy*
 - Negação: $\langle \neg A \rangle$
 - Conjunção: $\langle A \wedge B \rangle$
 - Disjunção: $\langle A \vee B \rangle$

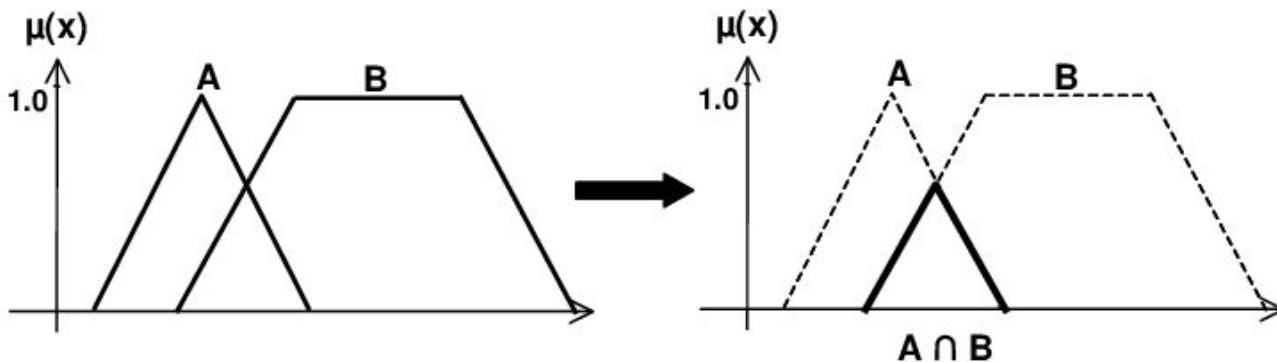
Lógica *fuzzy*

- Operadores básicos
 - Negação $\neg A$



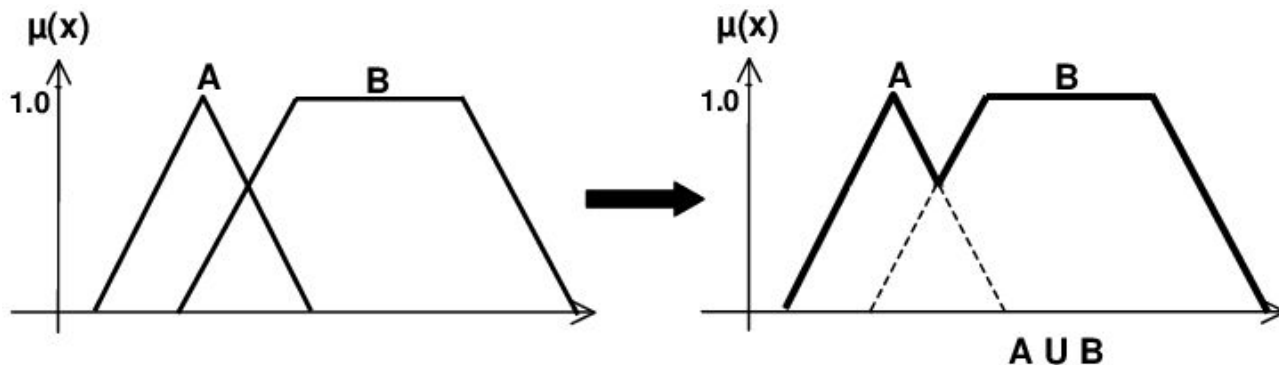
Lógica *fuzzy*

- Operadores básicos (cont.)
 - Conjunção $\langle A \wedge B \rangle$



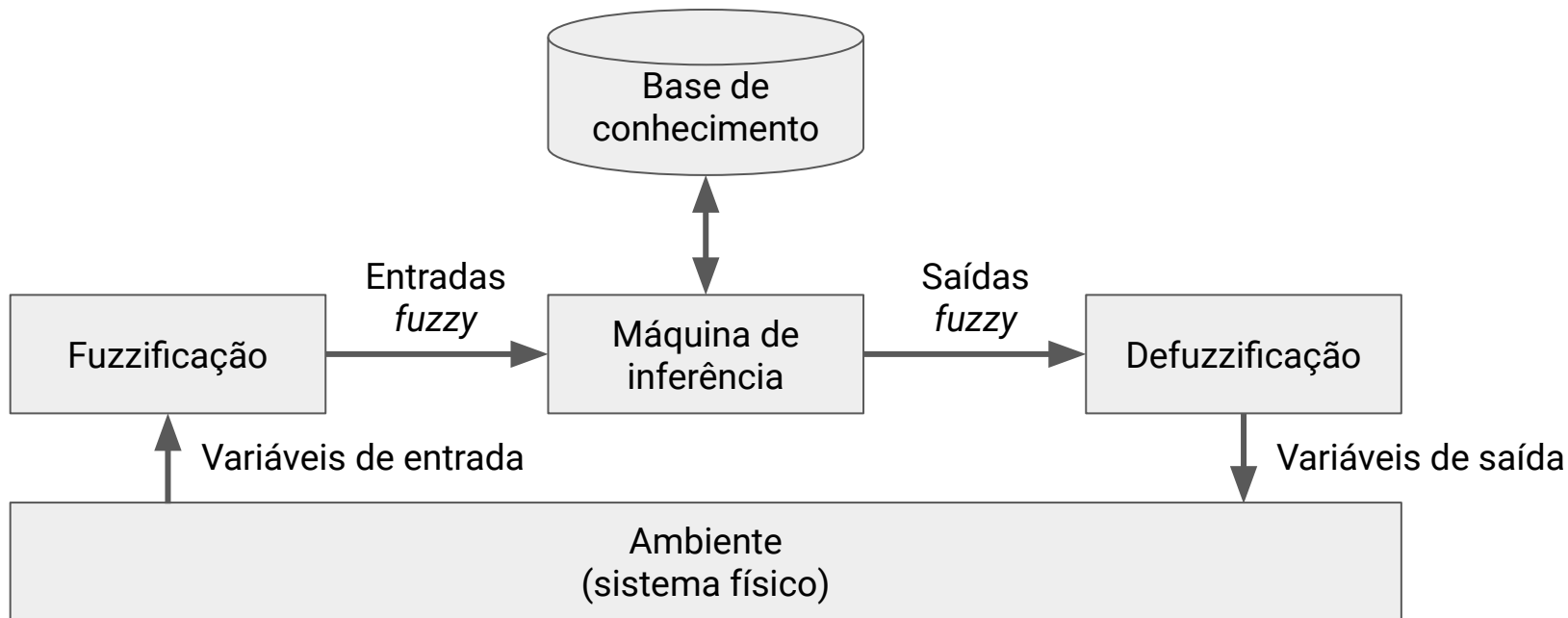
Lógica *fuzzy*

- Operadores básicos (cont.)
 - Disjunção $\langle A \vee B \rangle$



Sistemas de controle *fuzzy*

- Sistema de controle *fuzzy* baseado no modelo de Mamdani



Sistemas de controle *fuzzy*

- Componentes de um sistema de controle *fuzzy*
 - Variáveis *fuzzy* de entrada e de saída
 - Regras *fuzzy*
 - Fuzzificação e defuzzificação
- Variáveis *fuzzy* de entrada e de saída
 - Representação das entradas e saídas mapeadas por conjuntos *fuzzy*
 - A definição das variáveis *fuzzy* englobam
 - Análise do problema
 - Definição das variáveis linguísticas
 - Definição das funções de pertinência
 - Definição das regiões/rótulos

Sistemas de controle *fuzzy*

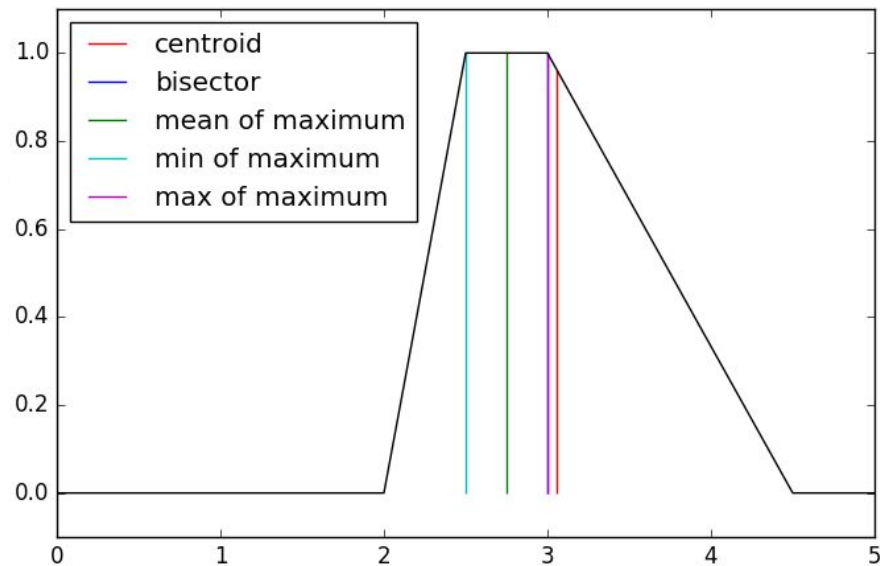
- Regras *fuzzy*
 - Regras de implicação usando variáveis *fuzzy*, cujo raciocínio consiste em
 - Avaliar o antecedente
 - Aplicar o resultado consequente
 - Disparo das regras
 - O grau de pertinência estabelece o grau de ativação de uma dada regra
 - Ou seja, quanto maior a compatibilidade entre a entrada e o antecedente da regra, maior será o peso do consequente no resultado final
 - Exemplo
 - **se** marcha > 5 **então** velocidade é 80 km/h (não *fuzzy*)
 - **se** marcha é alta **então** velocidade é alta (*fuzzy*)

Sistemas de controle *fuzzy*

- Fuzzificação e defuzzificação
 - Fuzzificação
 - Etapa onde as entradas do sistema são modeladas por conjuntos *fuzzy*
 - Transformação de entradas não *fuzzy*
 - As entradas representam valores numéricos capturados do ambiente
 - Defuzzificação
 - Interpretação das saídas *fuzzy* para saídas não *fuzzy*
 - Saídas não *fuzzy* representam comandos a serem executados no ambiente
 - Requer a definição do método a ser utilizado para realizar esta interpretação

Sistemas de controle *fuzzy*

- Métodos de defuzzificação
 - Centróide
 - Centro de gravidade do conjunto *fuzzy*
 - Mínimo ou maior máximo
 - Valor para o qual o grau de pertinência no conjunto é máximo
 - Média dos máximos
 - Média dos maiores valores da função de pertinência nos valores de saída *fuzzy*



Sistemas de controle *fuzzy*

- Ferramentas para desenvolvimento de sistemas de controle *fuzzy*
 - jFuzzyLogic
 - Biblioteca em Java para construção de sistemas *fuzzy*
 - Projeto *open-source*
 - FuzzyCLIPS
 - Extensão do CLIPS para possibilitar o uso da lógica *fuzzy*
 - Projeto *open-source*
 - Scikit-Fuzzy
 - Coleção de algoritmos fuzzy desenvolvidos em Python
 - Projeto *open-source*

Exercícios

2. Implemente um sistema fuzzy para calcular o percentual de gorjeta em um restaurante.

O cálculo da gorjeta deve levar em consideração as seguintes regras:

- Se a refeição estiver insossa e o serviço ruim, a gorjeta será pouca
- Se a refeição estiver saborosa e o serviço excelente, a gorjeta será generosa
- Se o tempo de atendimento for demorado, não haverá gorjeta
- Se o tempo de atendimento for mediano ou rápido, haverá gorjeta

Sistemas inteligentes

Inteligência artificial

Prof. Allan Rodrigo Leite