



EA 072 Inteligência Artificial em Aplicações Industriais

# 2-Agentes Inteligentes

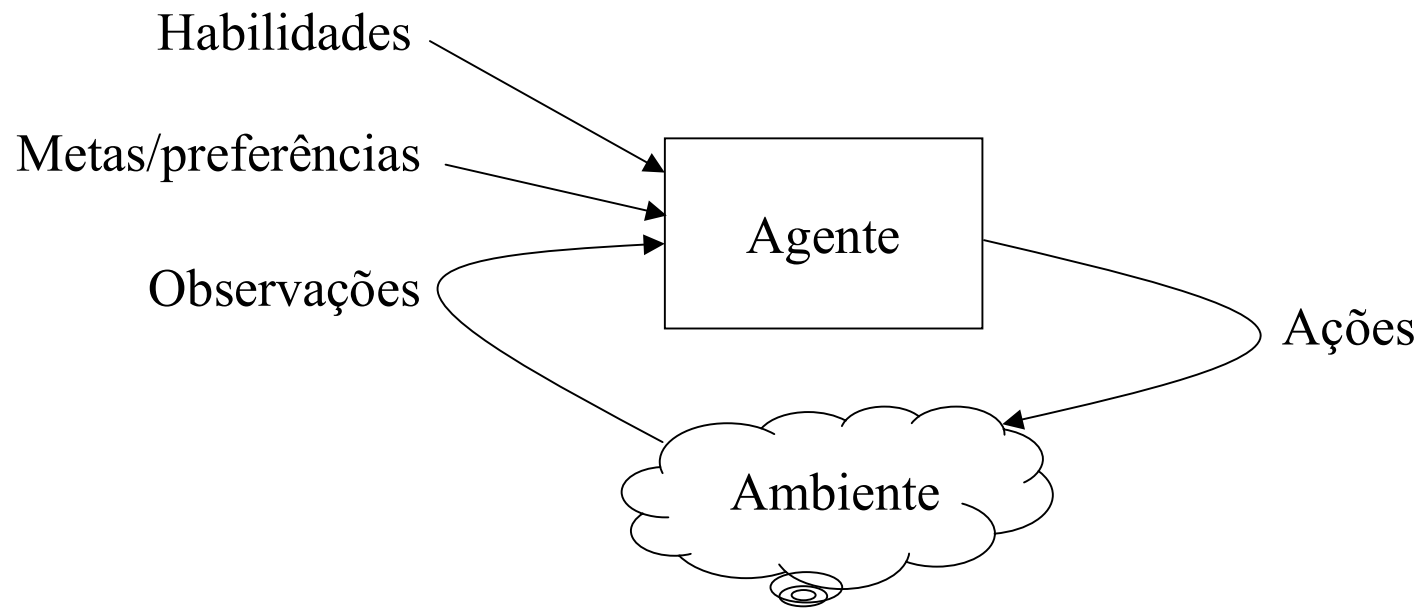
# Agentes e ambientes

- Agente

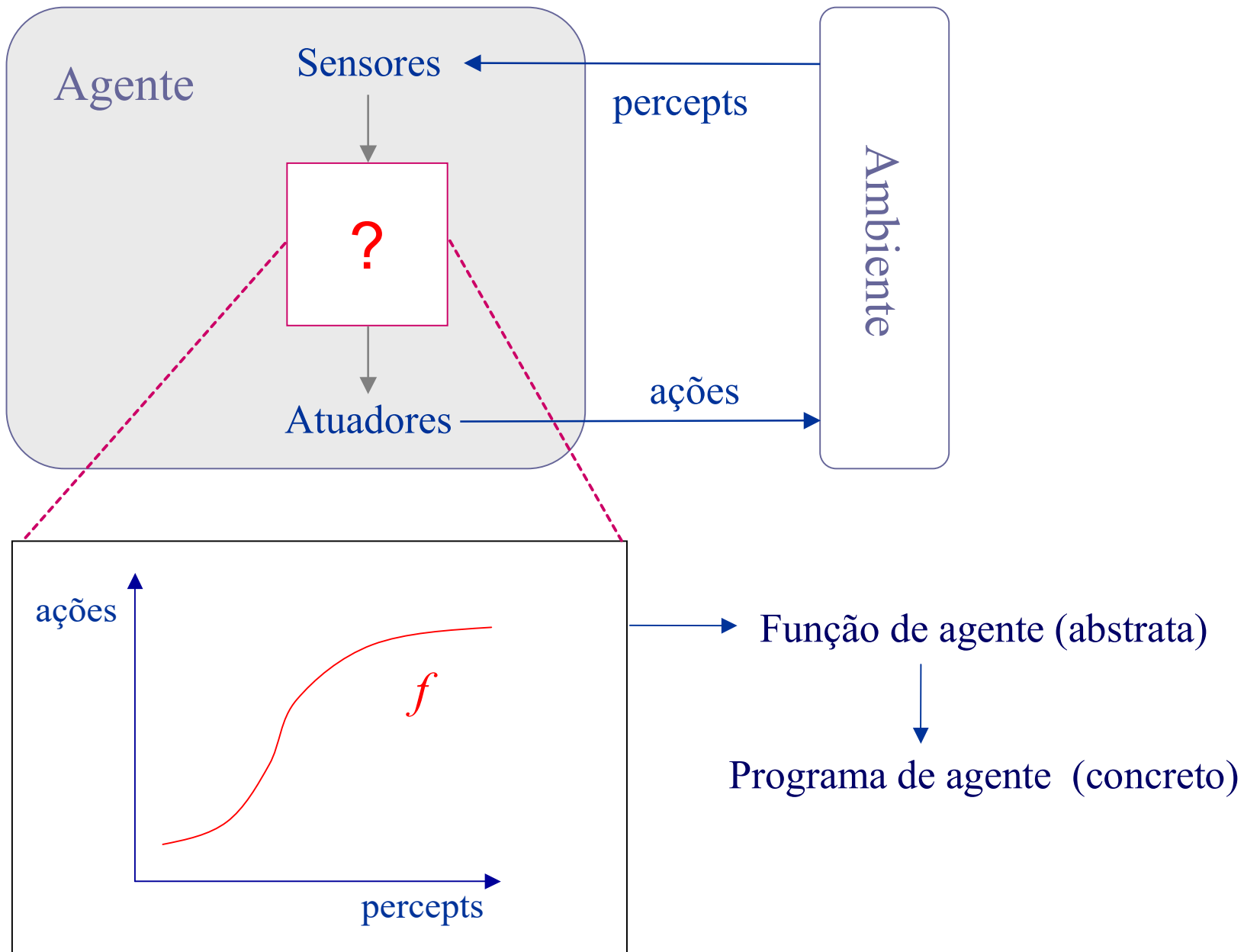
qualquer entidade que perceba seu ambiente através de sensores e atue sobre o ambiente através de atuadores. A noção de agente é uma ferramenta para analisar sistemas. (Russel & Norvig 2010)

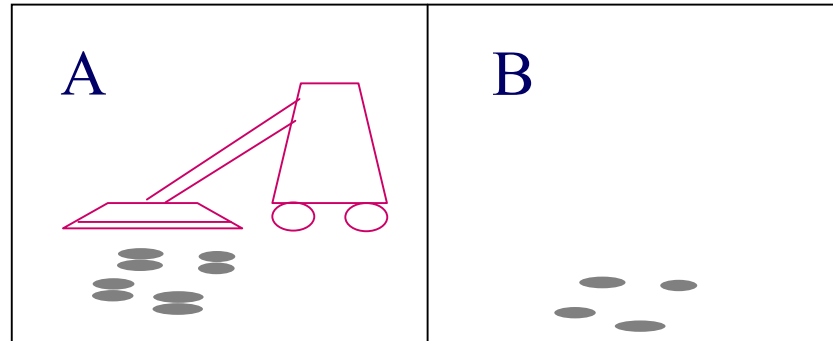
- Exemplos

- robôs
- web crawler
- veículos autônomos
- Siri
- controlador industrial
- etc.



- percept: uma observação, uma entrada em um instante
- história perceptiva: sequência completa de todas observações
- ações: saídas especificadas por um mapeamento de uma sequência perceptiva em uma ação





Sequência de percepts	Ação
$[A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Dirty}]$	<i>Suck</i>
$[B, \textit{Clean}]$	<i>Left</i>
$[B, \textit{Dirty}]$	<i>Suck</i>
$[A, \textit{Clean}], [A, \textit{Clean}]$	<i>Righ</i>
$[A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
:	:
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	<i>Righ</i>
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Such</i>
:	:

# Agente racional

- Agente racional é um conceito central na IA
  - conceito útil em uma variedade de aplicações/ambientes
  - auxilia o desenvolvimento de princípios de projeto
  - ajuda a construir agentes com chances de sucesso
  - agentes (razoavelmente) inteligentes
- Racionalidade depende
  - medida de desempenho que define o critério de sucesso
  - história perceptiva
  - o que *a priori* o agente sabe sobre o ambiente
  - as ações que o agente pode executar

# Agente racional

Para cada sequência perceptiva possível, um agente racional deve selecionar a ação que maximize seu critério de desempenho, dada a evidência proporcionada pela sequência perceptiva e qualquer outro conhecimento que o agente tenha disponível.

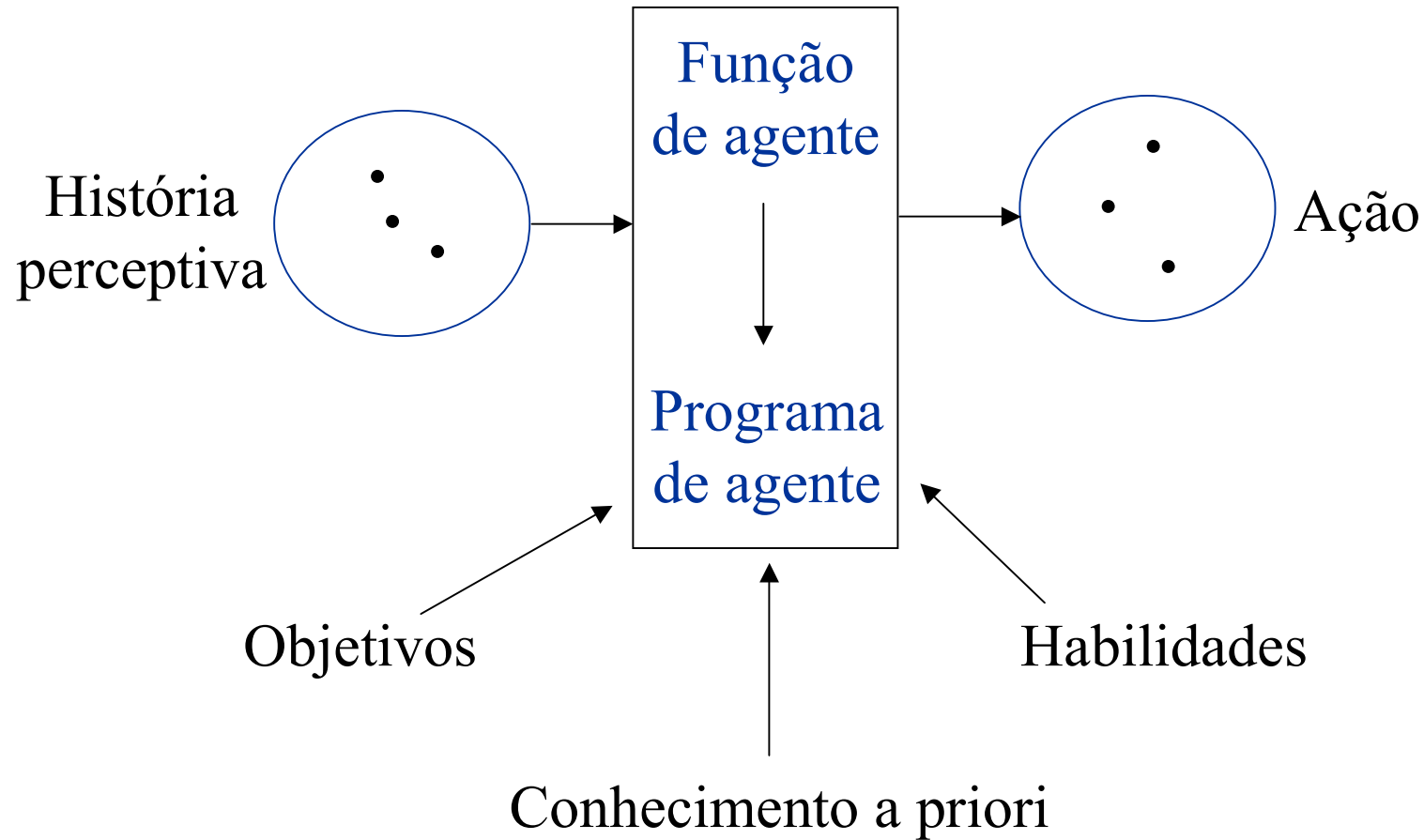
Agente racional é aquele que faz a coisa certa, se comporta da melhor maneira possível em um ambiente.

Melhor desempenho caracterizado pela medida de desempenho que avalia qualquer sequência de estados do ambiente.

- **Mapeamentos definem agentes racionais:** a especificação das ações a serem realizadas pelos agentes em resposta a uma sequência perceptiva define um *projeto* de um agente racional.
- A racionalidade maximiza o desempenho esperado, enquanto que a perfeição maximiza o desempenho real (Russel & Norvig, 2010)
- Onisciência, aprendizagem e autonomia
- Racional  $\neq$  onisciente
- Racional: autonomia + aprendizagem



# Projeto de agentes



# Projeto de agentes

- Demanda uma idéia sólida a respeito de:
  - percepções e ações
  - critérios de desempenho e/ou metas a serem atingidas
  - natureza do ambiente
- **PEAS: Performance, Environment, Actuators, Sensors**
- Real × artificial: robôs × *softbots* (agentes de software)

## ■ Natureza do ambiente

- observável × parcialmente observável
- determinístico × estocástico
- episódico × sequencial
- estático × dinâmico
- discreto × contínuo
- conhecido × desconhecido

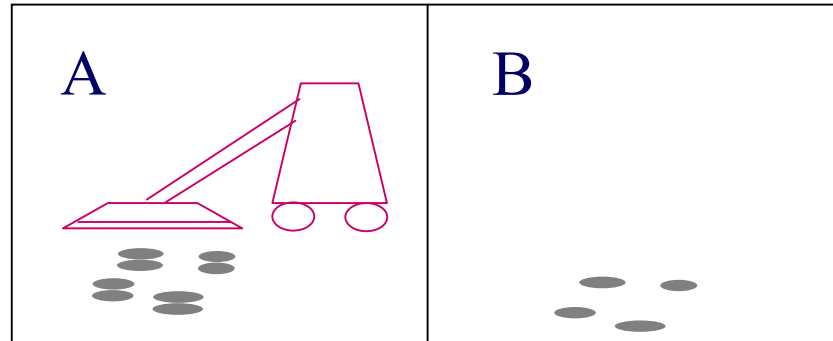
# Exemplos de agentes (PEAS)

TIPO AGENTE	PERFORMANCE	ENVIRONMENT	ACTUATORS	SENSORS
Diagnóstico médico	Saúde do paciente, custos	Paciente, hospital, pessoal	Perguntas, testes, exames, tratamentos	Sintomas, respostas do paciente
Análise imagens de satélites	Categorização correta	Downlink do satélite em órbita	Display de uma categorização da cena	Pixels de intensidade e cor variada
Robô de manuseio	Percentagem de peças no lugar correto	Correia transportadora com peças	Juntas com braço e mão	Câmera, sensores de posição angular
Controle de processos	Maximizar pureza, produção, segurança	Refinaria, operadores	Válvulas, bombas, aquecedores	Temperatura, pressão, vazão, químicos
Tutor de línguas	Maximizar nota dos testes	Conjunto de estudantes em teste	Display de exercícios, sugestões, correções	Entrada por teclado

# Estrutura de agentes

- Tarefa da IA: projetar um programa de agente que
  - implementa a função de agente
  - seja executável em uma arquitetura
  - arquitetura = dispositivo computação + sensores + atuadores

**Agente = Arquitetura + Programa**



Sequência de percepts	Ação
$[A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Dirty}]$	<i>Suck</i>
$[B, \textit{Clean}]$	<i>Left</i>
$[B, \textit{Dirty}]$	<i>Suck</i>
$[A, \textit{Clean}], [A, \textit{Clean}]$	<i>Righ</i>
$[A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
:	:
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	<i>Righ</i>
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Such</i>
:	:

## Programa de agente: exemplo

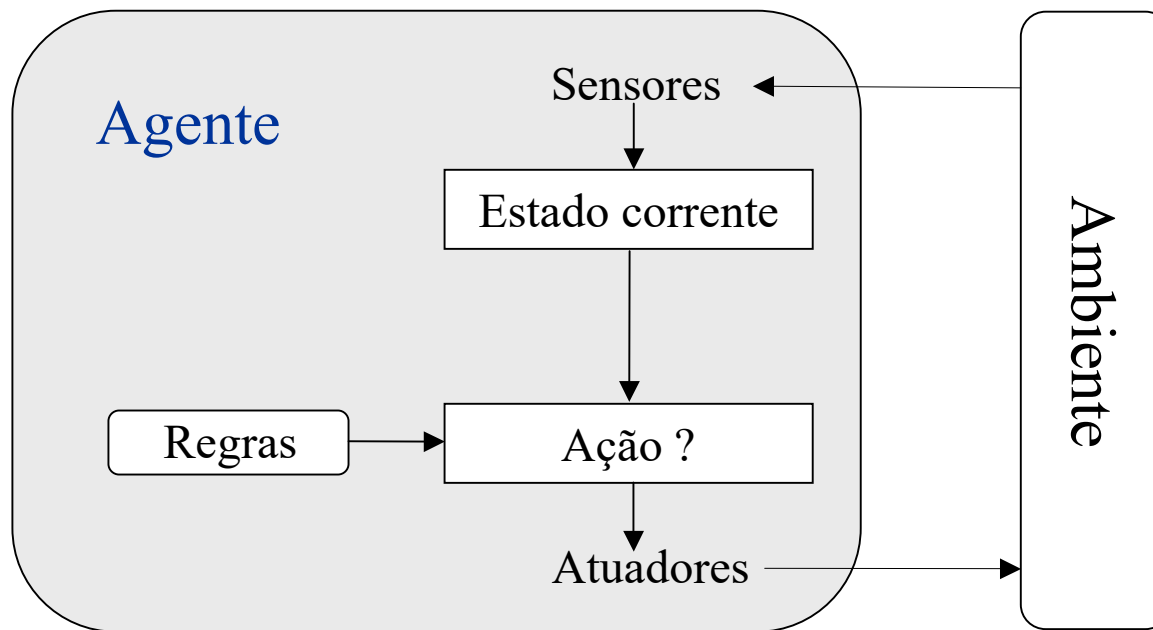
**function** TABLE\_DRIVEN\_AGENT(*percept*) **returns** an action  
**persistent:** *percepts*, sequencia perceptiva, inicialmente vazia  
*table*, tabela de ações indexada por *percepts*, inicia  
especificada completamente

append *percept* to the end of *percepts*

*action* ← LOOKUP(*percepts*, *table*)

**return** *action*

# Agente reativo

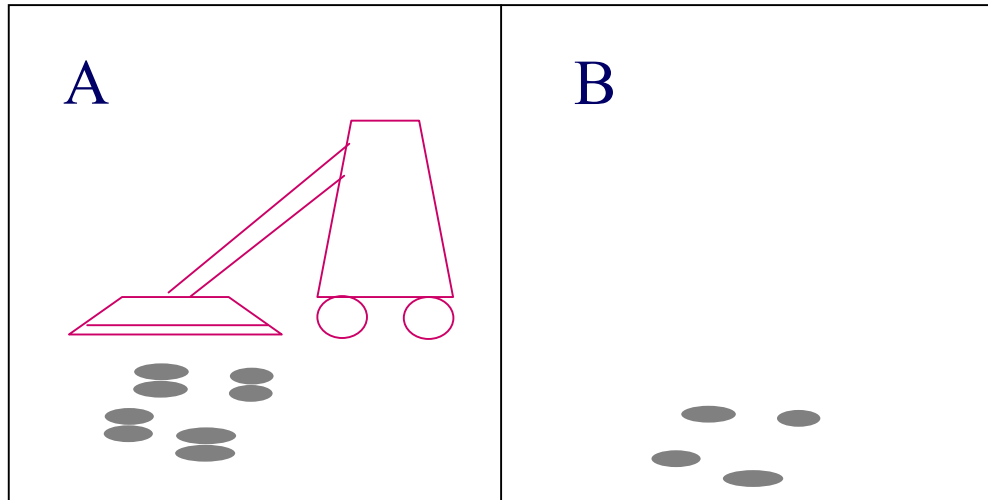




# Programa de agente reativo

**function** SIMPLE\_REFLEX\_AGENT (*percept*) **returns** an action  
**persistent:** *rules*, conjunto de regras condição-ação

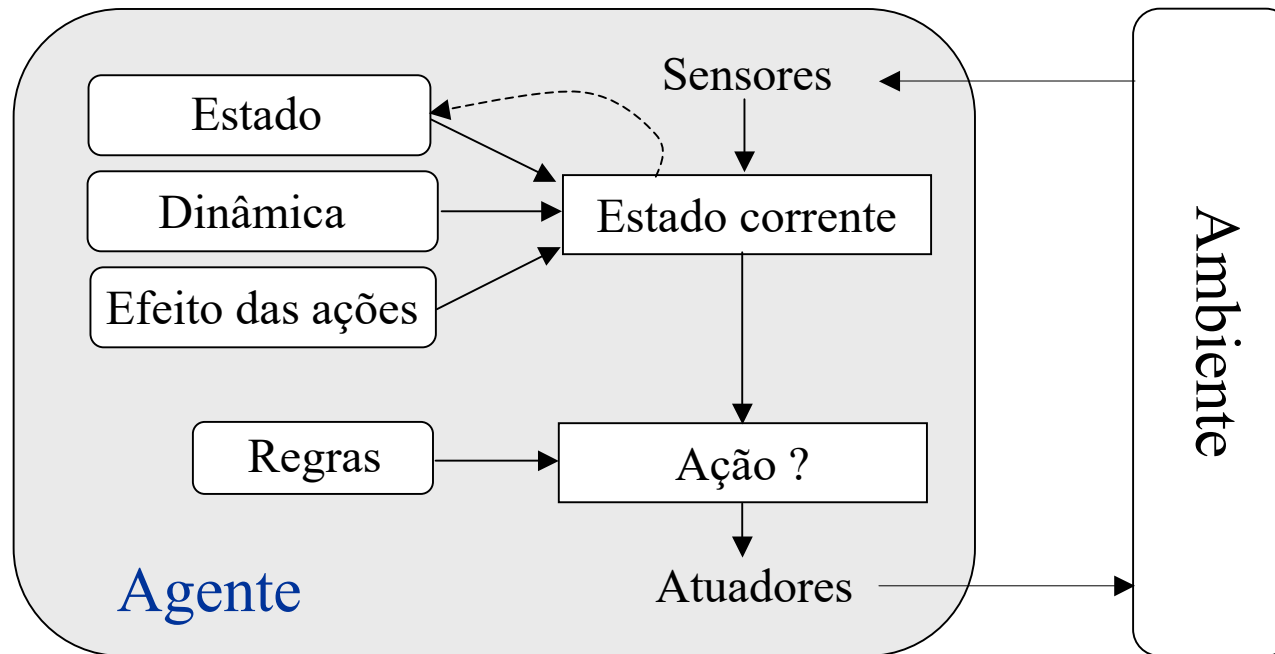
```
state ← INTERPRET_INPUT(percept)  
rule ← RULE_MATCH(state, rules)  
action ← rule.ACTION  
return action
```



**function** REFLEX-VACUUM-AGENT (*[location, status]*) **returns** an action

**if** *status = Dirty* **then return** *Suck*  
**else if** *location = A* **then return** *Right*  
**else if** *location = B* **then return** *Left*

# Agente reativo com modelo (estado interno)



# Programa de agente reativo com modelo

**function** MODEL\_BASED\_REFLEX\_AGENT (*percept*) **returns** an action

**persistent:** *state*, descrição do estado corrente

*model*, descreve como o estado evolui

*rules*, conjunto de regras condição-ação

*action*, ação mais recente, inicialmente nenhuma

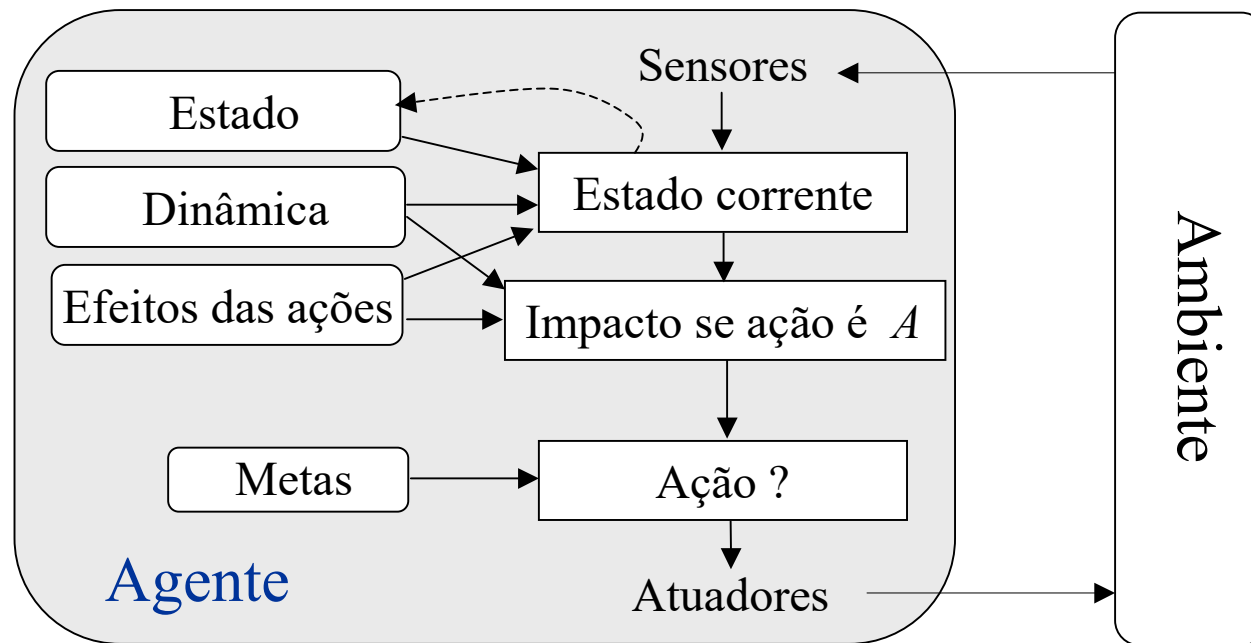
*state* ← UPDATE\_STATE(*state*, *action*, *percept*, *model*)

*rule* ← RULE\_MATCH(*state*, *rules*)

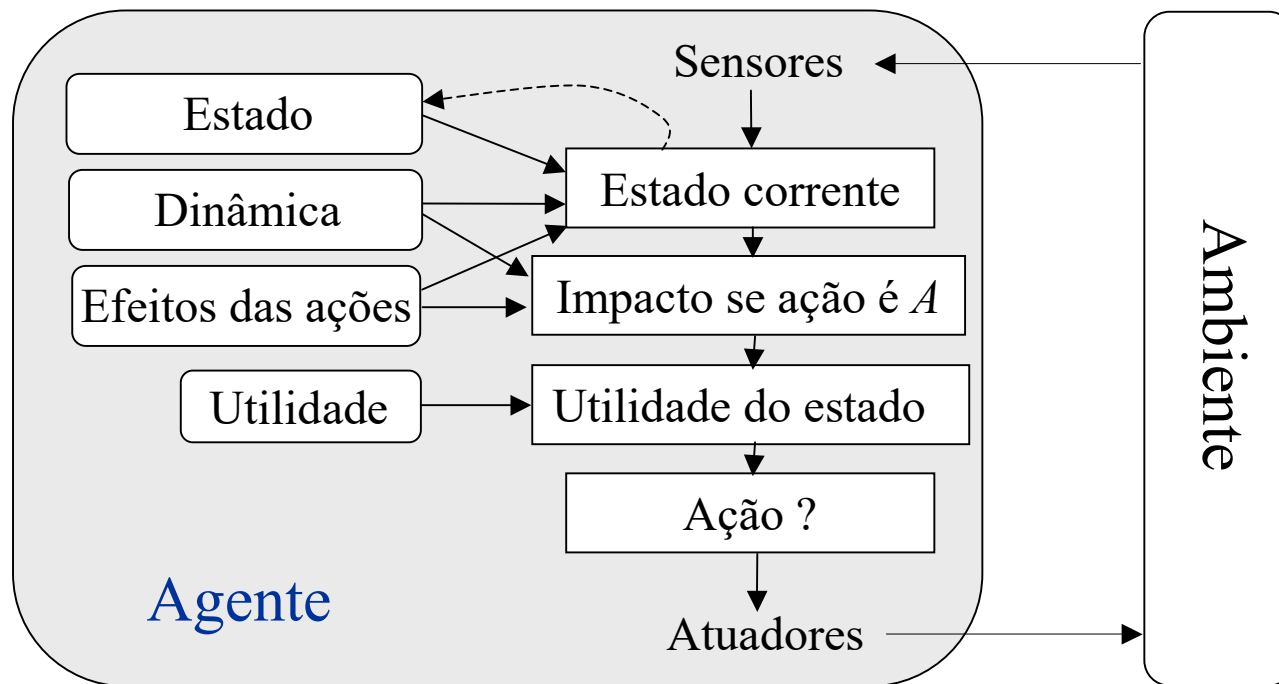
*action* ← *rule*.ACTION

**return** *action*

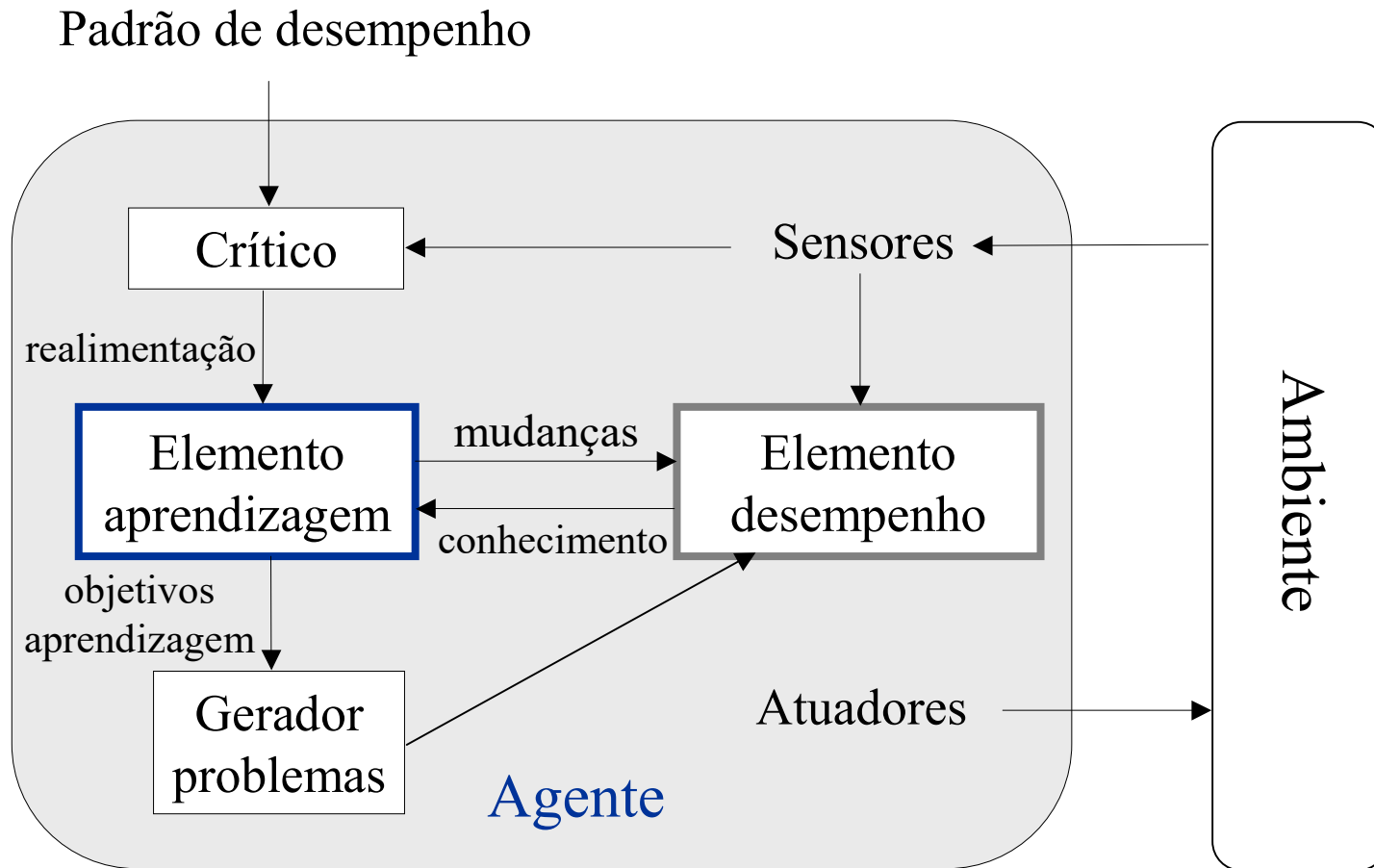
# Agente orientado por metas



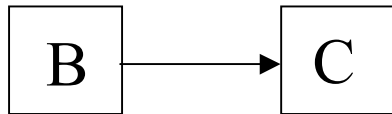
# Agente orientado por utilidade



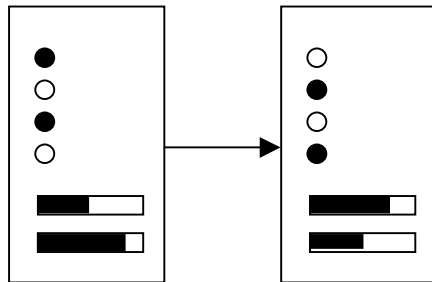
# Agente de aprendizagem



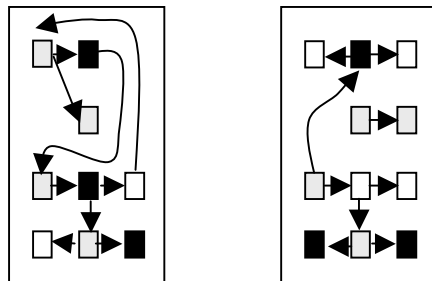
# Representação de componentes de agentes



Atômica: sem estrutura interna  
e.g. busca, jogos, HHM



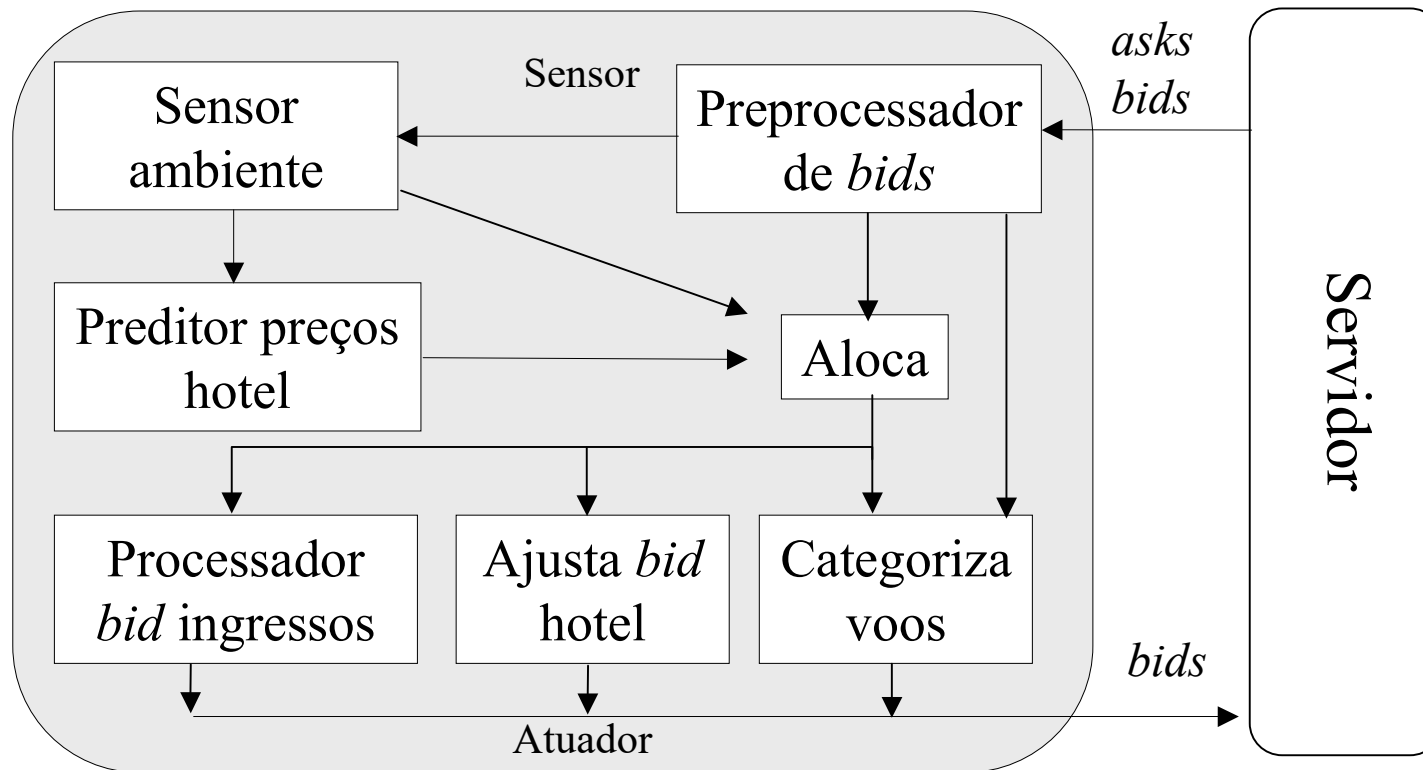
Fatorada: variável, atributo, valor  
e.g. Bayes net, CS, LP, ML



Estruturada: objetos, relações  
e.g. LPO,DB, KBL, NL



# Fuzzy trading agent



Fonte: He, M., Jennings, R., Designing a successful trading agent: A fuzzy set approach, *IEEE Trans. on Fuzzy Systems*, vol.12, no. 3, pp. 389-410, 2004.

## Observação

Este material refere-se às notas de aula do curso EA 072 Inteligência Artificial em Aplicações Industriais da Faculdade de Engenharia Elétrica e de Computação da Unicamp. Não substitui o livro texto, as referências recomendadas e nem as aulas expositivas. Este material não pode ser reproduzido sem autorização prévia dos autores. Quando autorizado, seu uso é exclusivo para atividades de ensino e pesquisa em instituições sem fins lucrativos.