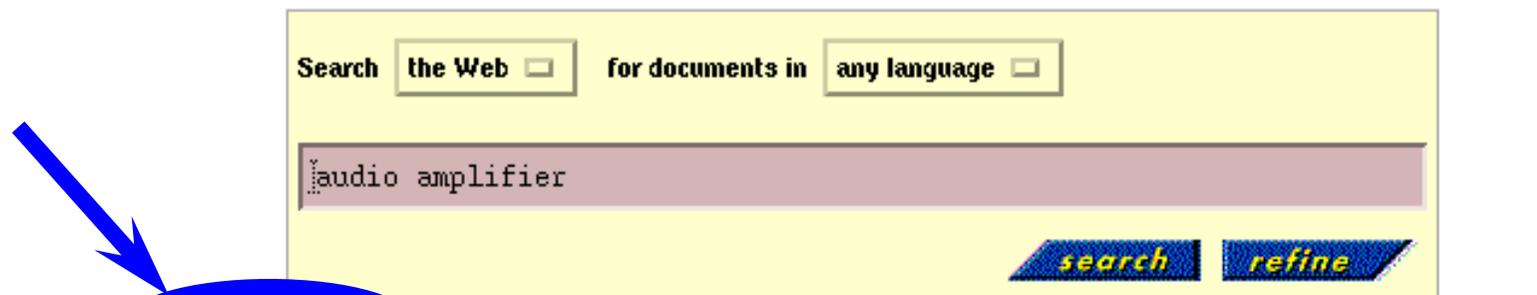


Agentes Inteligentes

Inteligência Artificial PCS3438

*Anna Helena Reali Costa
Escola Politécnica da USP
Engenharia de Computação (PCS)*

Problema: Auxílio a Compras na Web



Search for documents in

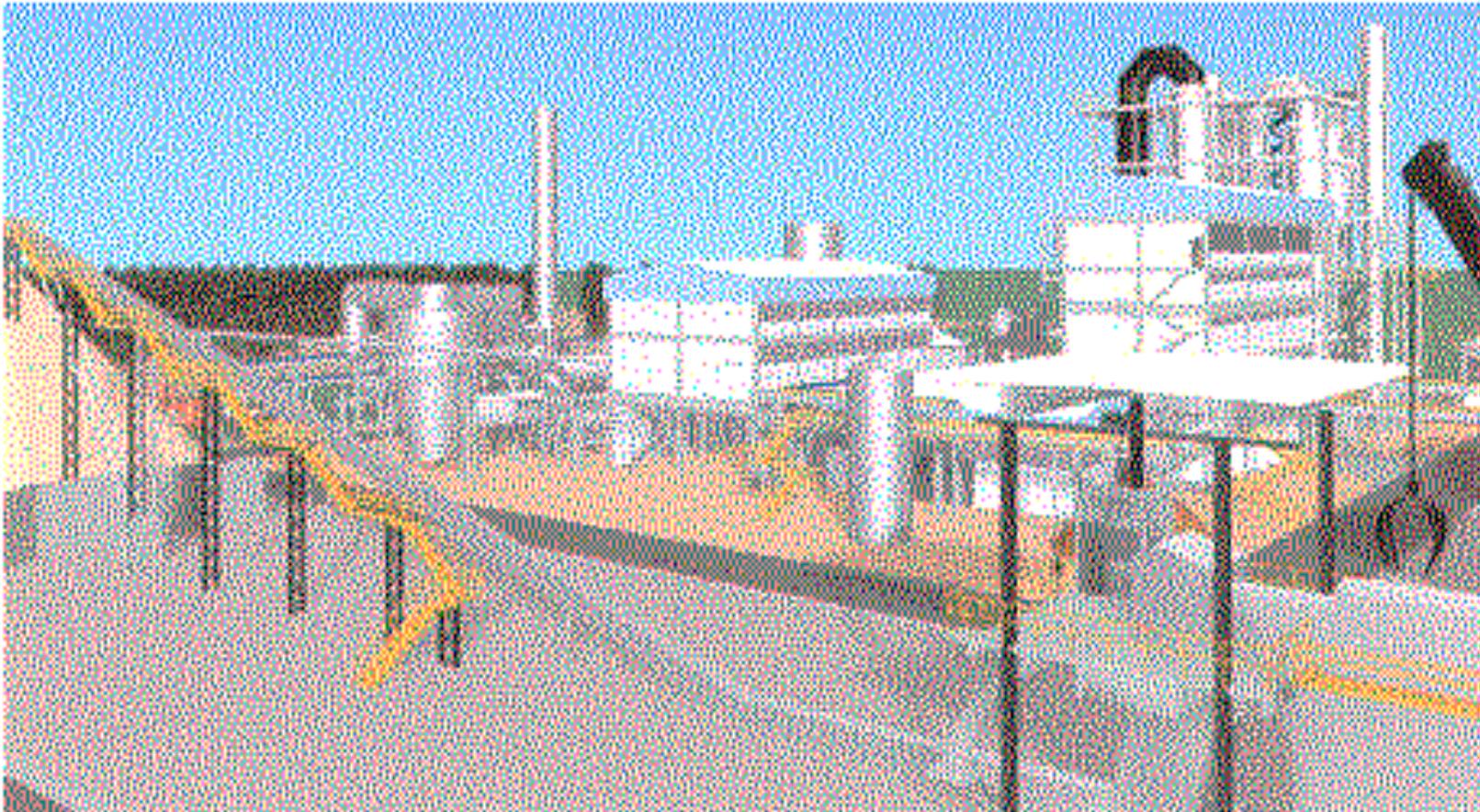
[Help](#) . [Preferences](#) . [New Search](#) . [Advanced Search](#)

[Click to find related books at Amazon.com.](#)

About 11002 documents match your query.

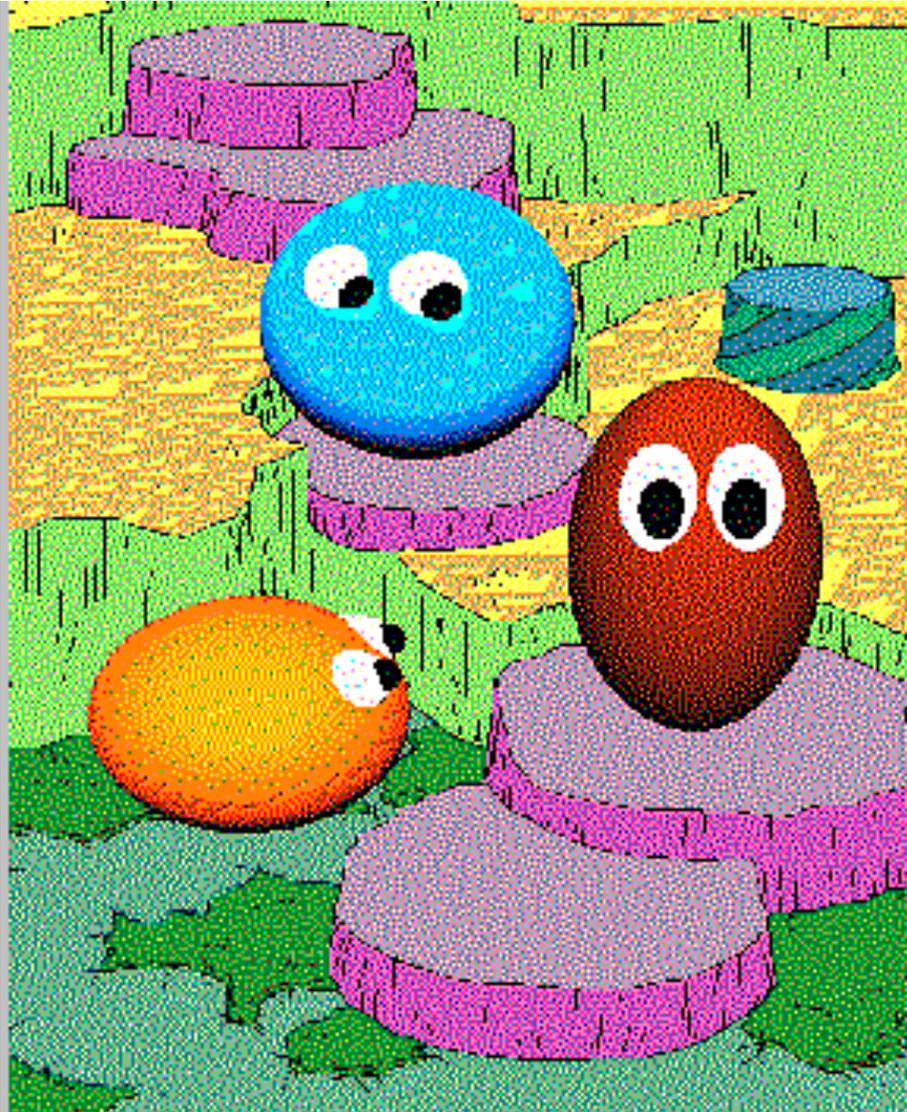
- [SIP326BC Stereo Audio Amplifier](#)**
Presents: STEREO AUDIO AMPLIFIER. SIP326BC. SCOPE OF THE STUDY. The stereo audio amplifier panel contains 12 fault insertion switches used to teach...
<http://www.omgsc.com/sip326bc.htm> - size 5K - 23-Oct-96 - English
- [Audio Amplifier Fine Tuning](#)**
Audio Amplifier Fine Tuning. Most ham CW and SSB communication is between 300HZ to 3000HZ. The following is a chart reprinted from the SGS data sheet...
<http://www.pan-tex.net/ust/r/receivers/ra01033.htm> - size 1K - 27-May-97 - English
- [Audio Amplifier Users](#)**
SB Electronics, Inc. Specialists in Film/Foil Capacitor Design and Manufacturing Since 1959! Audio Amplifier Users. Don't settle for anything but the...
<http://www.sbelectronics.com/user.htm> - size 15K - 8-Aug-97 - English
- [Detail: CK151 Audio Amplifier Kit](#)**
CK151 Audio Amplifier Kit. \$16.90. A universal audio amplifier with many applications. Includes a microphone jack. Microphone input sensitivity: 5 mV...
<http://www.shcpsite.com/kits/prod67.html> - size 2K - 19-Aug-97 - English
- [Audio Amplifier](#)**
Audio Amplifier. There have been two audio amplifiers that I have used for this receiver. One using a TDA2002, built exactly like the one in the Beginner..
<http://www.pan-tex.net/ust/r/receivers/audio.htm> - size 2K - 26-May-97 - English

Problema: Automação de sistemas de potência (*Smart Grids*)



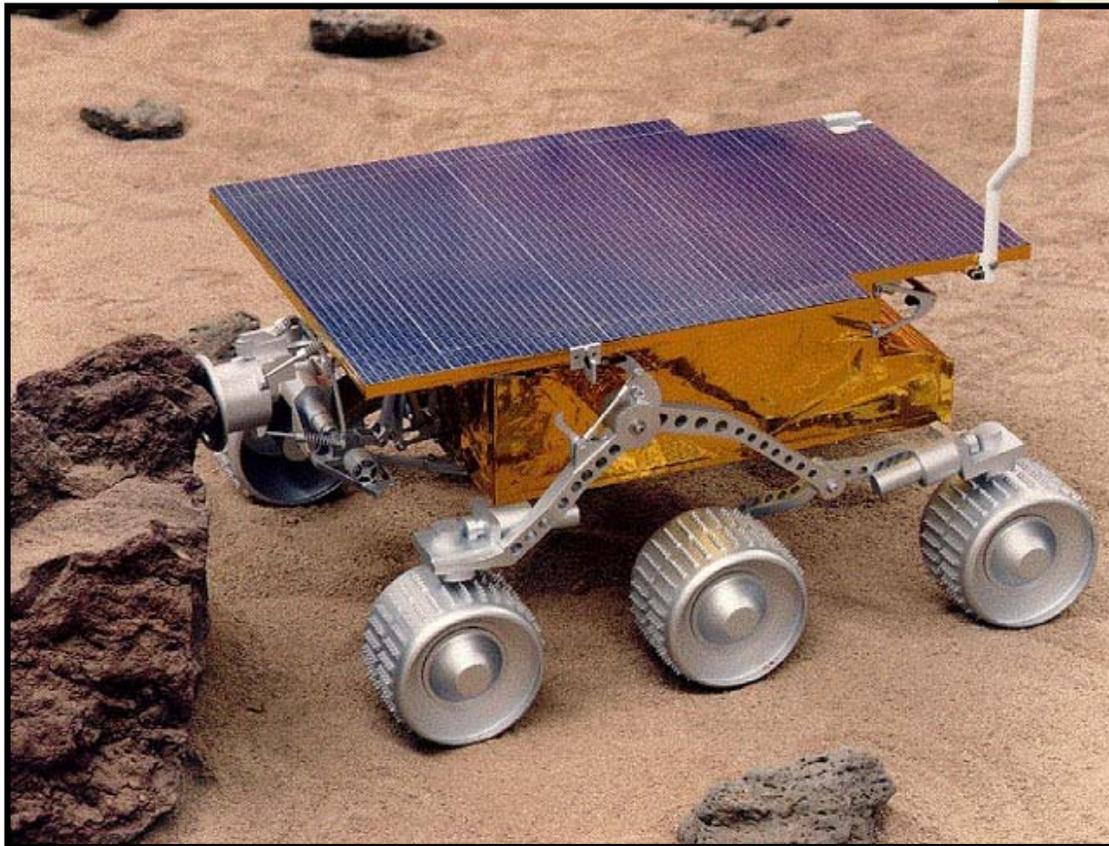
objetos: rios, barragens, turbinas, transformadores, linhas, ...

Problema: Produção de histórias interativas



- . Criar ilusão da vida (ex. Walt Disney)
- . Permitir interação com usuário
- . Modelar comportamento e personalidade (ex. tamagotchi)

Problema: exploração planetária



Problema: robôs de estimação



QRIO - SONY

AIBO - SONY



NAO - RoboLab

ASIMO - HONDA



E aí?

- O que estes problemas têm em comum?
 - Grande complexidade (número, variedade e natureza das tarefas)
 - Não há “solução algorítmica”, mas existe conhecimento
 - Modelagem do comportamento de um ser inteligente (autonomia, aprendizagem, conhecimento, etc.)

O que é “ser inteligente”?

- Ser inteligente é atuar como humanos.
- Ser inteligente é “pensar” como humanos.
- Ser inteligente é “pensar” racionalmente.
- **Ser inteligente é atuar racionalmente.**

Racionalidade = capacidade de alcançar o sucesso esperado na execução de uma tarefa.

Grau de Sucesso: medida de desempenho a ser maximizada.

Um programa de IA pode ser visto como um Agente Racional

- Plano do tópico
 - O que é um Agente Racional (inteligente)?
 - Ambientes e Arquiteturas
 - Aplicações
 - Estado atual do conceito de agente

O que é um agente?

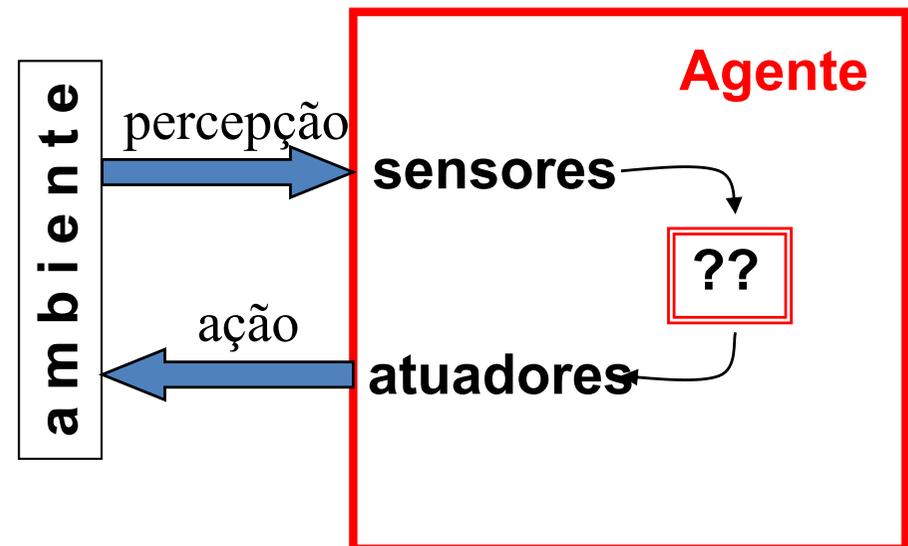
- Agente é qualquer entidade que:
 - **percebe** seu ambiente através de sensores (ex. câmeras, microfone, teclado, *finger*, ...)
 - **age** sobre ele através de atuadores (ex. vídeo, auto-falante, impressora, braços, ftp, ...)

■ Ambiente/agente

- Físico: robôs
- Software: softbots
- Realidade virtual: avatares

■ Efetua o mapeamento:

- sequência perceptiva → ação



Medida de Desempenho (MD)

- Critério que define o grau de sucesso de um agente na realização de uma dada tarefa
 - O quê avaliar, Como avaliar, Quando avaliar
 - Esta medida deve ser imposta do exterior
 - Má escolha da MD pode acarretar comportamento indesejado
 - Compromissos entre objetivos múltiplos conflitantes
 - Resta o problema de saber **quando** avaliar o desempenho

Agente Racional

- Para cada sequência perceptual possível, o agente racional deve selecionar uma ação que ele **espera que maximize sua medida de desempenho**, segundo a evidência dada pela sequência perceptiva e seus eventuais conhecimentos.
 - Limitações de sensores, atuadores, “raciocinador” (conhecimento, tempo, etc.)
 - Agente racional deve ser **autônomo**:
 - Ter capacidade de raciocínio, decisão e de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência

A metáfora de agente decompõe:

1) Problema em:

- percepções, ações, objetivos e ambiente (e outros agentes)

2) Tipo de conhecimento em:

- Quais são as propriedades relevantes do mundo
- Como o mundo evolui
- Como identificar os estados desejáveis do mundo
- Como interpretar suas percepções
- Quais as conseqüências de suas ações no mundo
- Como medir o sucesso de suas ações
- Como avaliar seus próprios conhecimentos

Agente de polícia

Agente



raciocínio

Conhecimento:

- leis
- comportamento dos indivíduos,...

Objetivo:

- fazer com que as leis sejam respeitadas

Ações:

- multar
- apitar
- parar, ...

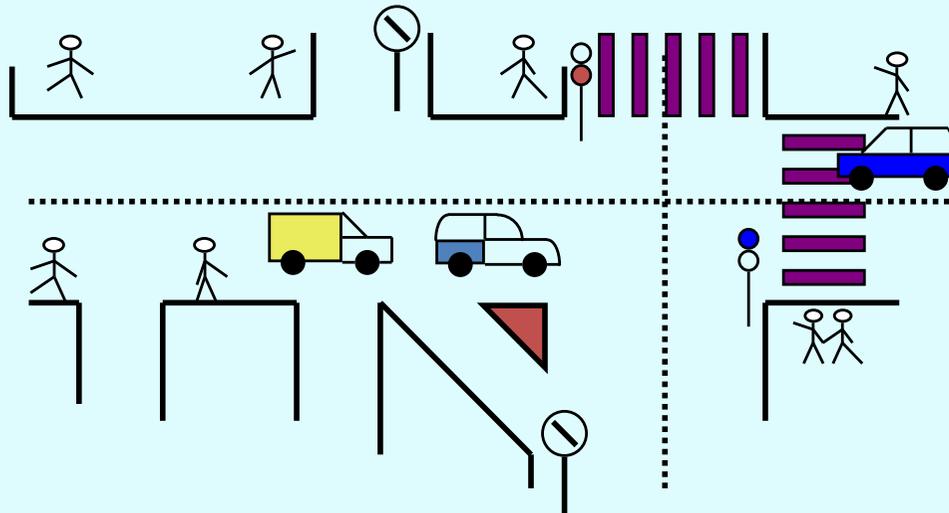
percepção



atuação



Ambiente



Propriedades do Ambiente

- totalmente observável x parcialmente observável
- determinístico x estocástico (envolve previsibilidade do próximo estado)
- episódico x sequencial (envolve ter ou não consequências futuras decorrentes da decisão atual)
- estático x dinâmico (envolve alterações fora do controle do agente)
- discreto x contínuo (aplicado a estado, tempo, ações e/ou percepções)
- único agente x multiagente (envolve comunicação, cooperação, competição..)

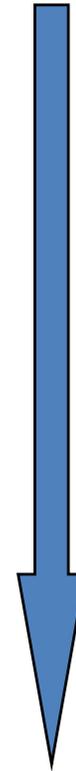
Estrutura do Agente

$$\text{Agente} = \text{arquitetura de HW} \\ + \\ \text{arquitetura de SW}$$

- Arquitetura de HW: qual é o dispositivo computacional, os sensores e atuadores
- Arquitetura de SW: módulos básicos do programa e suas interrelações

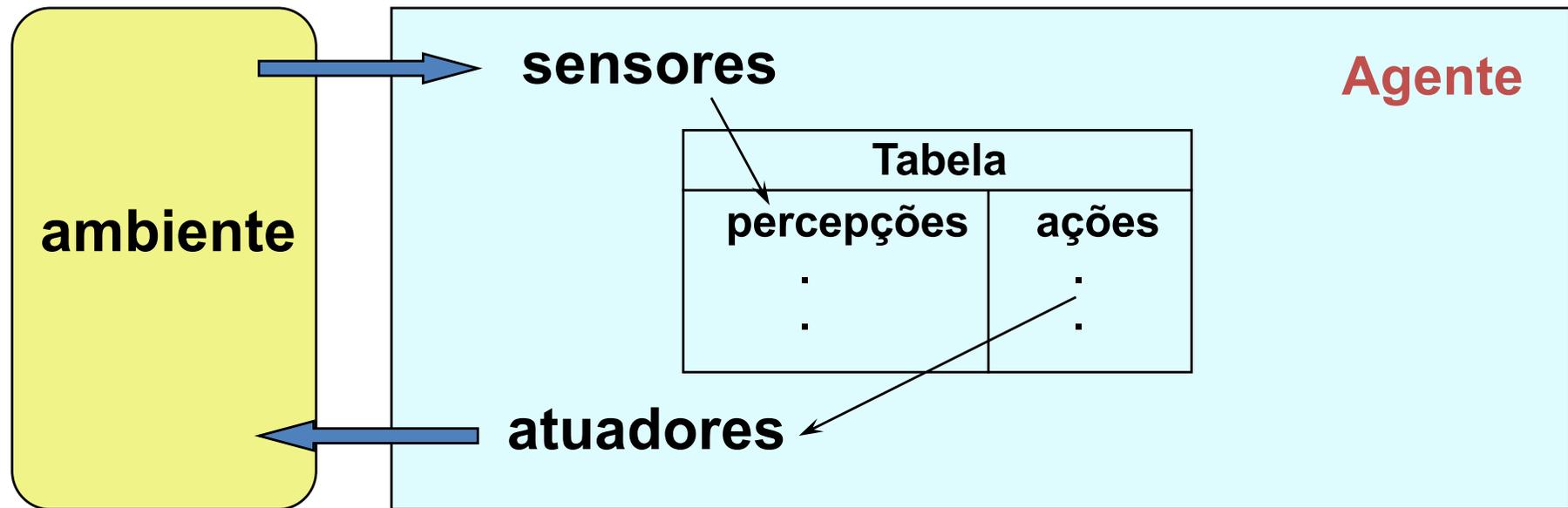
Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz



autonomia
complexidade

Agente tabela

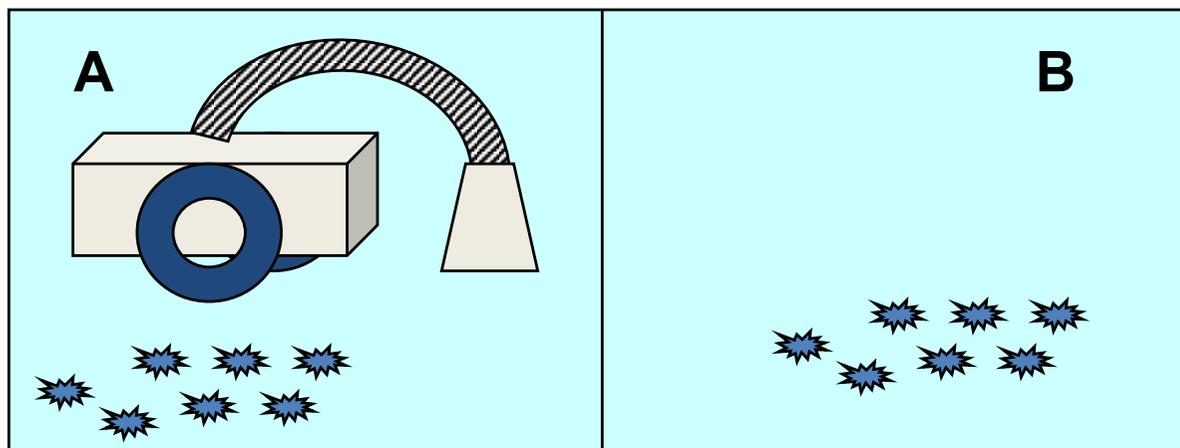


- **Limitações**

- Mesmo para problemas simples → tabelas muito grandes
 - ex. xadrez 30^{100}
- Nem sempre é possível, por ignorância ou questão de tempo, construir a tabela
- Não há autonomia nem flexibilidade

- **Ambientes**

- observável, determinístico, episódico, estático, discreto (e minúsculo!)



PERCEPÇÃO [sala, estado]	AÇÃO
[A, limpo]	Ir para a direita
[A, sujo]	Aspirar
[B, limpo]	Ir para a esquerda
[B, sujo]	Aspirar
[A, limpo], [A, limpo]	Ir para a direita
....
[A, limpo], [A, limpo], [A,sujo]	Aspirar
...	...

function Agente-Tabela-Aspirador (*percept*) **return** uma ação

static: *percepts* – uma seqüência, inicialmente vazia

table – uma tabela de ações indexada pela seqüência perceptiva, inicialmente totalmente especificada

append *percept* **ao final de** *percepts*

ação ← **Lookup** (*percepts*, *table*)

return *ação*

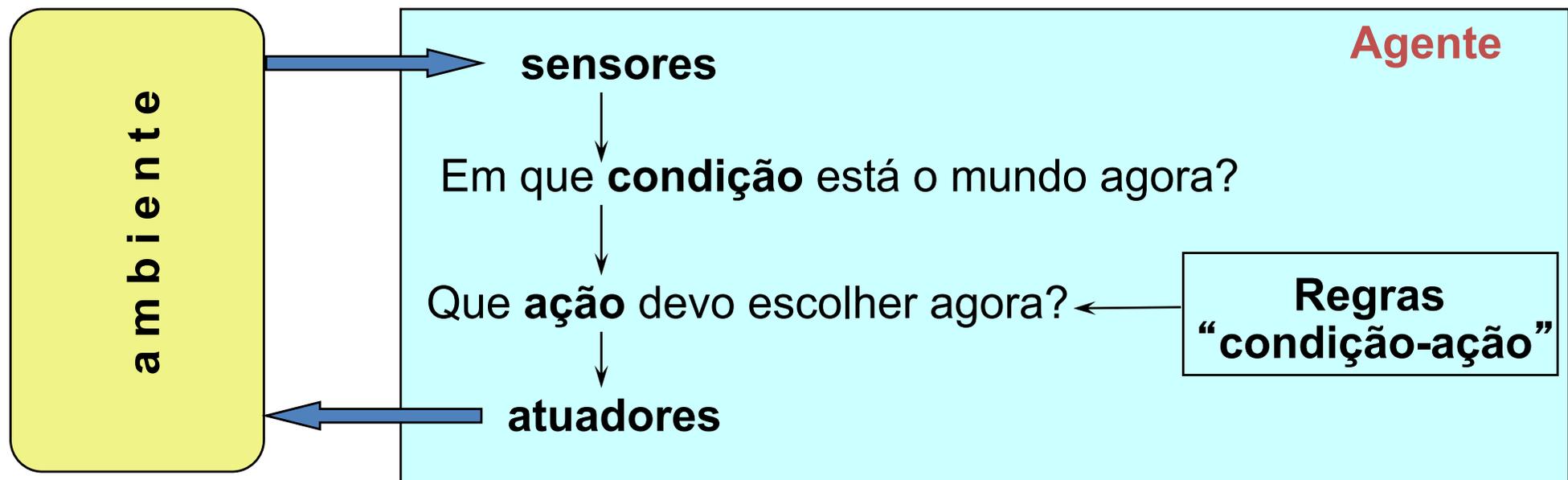
Uso MUITO limitado (impossível): seja P o conjunto de percepções possíveis e T , o tempo de vida do agente →
entradas da tabela = $\sum_{t=1..T} |P|^t$

→ **Como IA soluciona isso?**

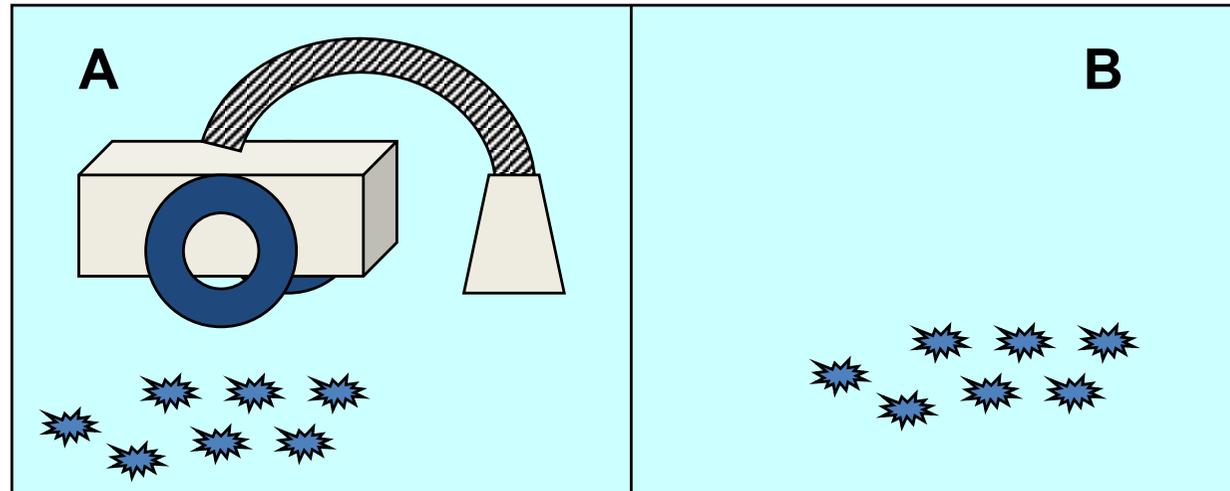
Arquiteturas

- Agente tabela
- **Agente reativo**
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz

Agente reativo



- Vantagens e desvantagens
 - Regras condição-ação: representação inteligível, modular e eficiente
 - ex. **Se** velocidade > 60 **então** multar
 - Não pode armazenar uma sequência perceptiva, pouca autonomia
- Ambientes:
 - Reflexo imprescindível em ambientes dinâmicos
 - Observável, episódico, pequeno



```
function Agente-Reativo-Aspirador ([local, status]) return uma ação  
  
if status=Sujo then return Aspira  
else if local=A then return Direita  
else if local=B then return Esquerda
```

OBS: utiliza a percepção corrente e ignora a história perceptual.

function Agente-Reflexo-Simples (*percept*) **return** uma ação

static: *regras* – um conjunto de regras condição-ação

estado ← **Interpreta-Entrada**(*percept*)

regra ← **Acha-Regra**(estado, *regras*)

ação ← **Regra-Ação** [regra]

return ação

Uso limitado: o ambiente tem que ser totalmente observável, pois o agente só funciona apropriadamente se a regra correta for disparada, o que depende da percepção atual realizada.

Arquiteturas

- Agente tabela
- Agente reativo
- **Agente baseado em modelo**
- Agente baseado em objetivos
- Agente baseado em utilidade
- Agente aprendiz

Necessidade de um modelo (1)

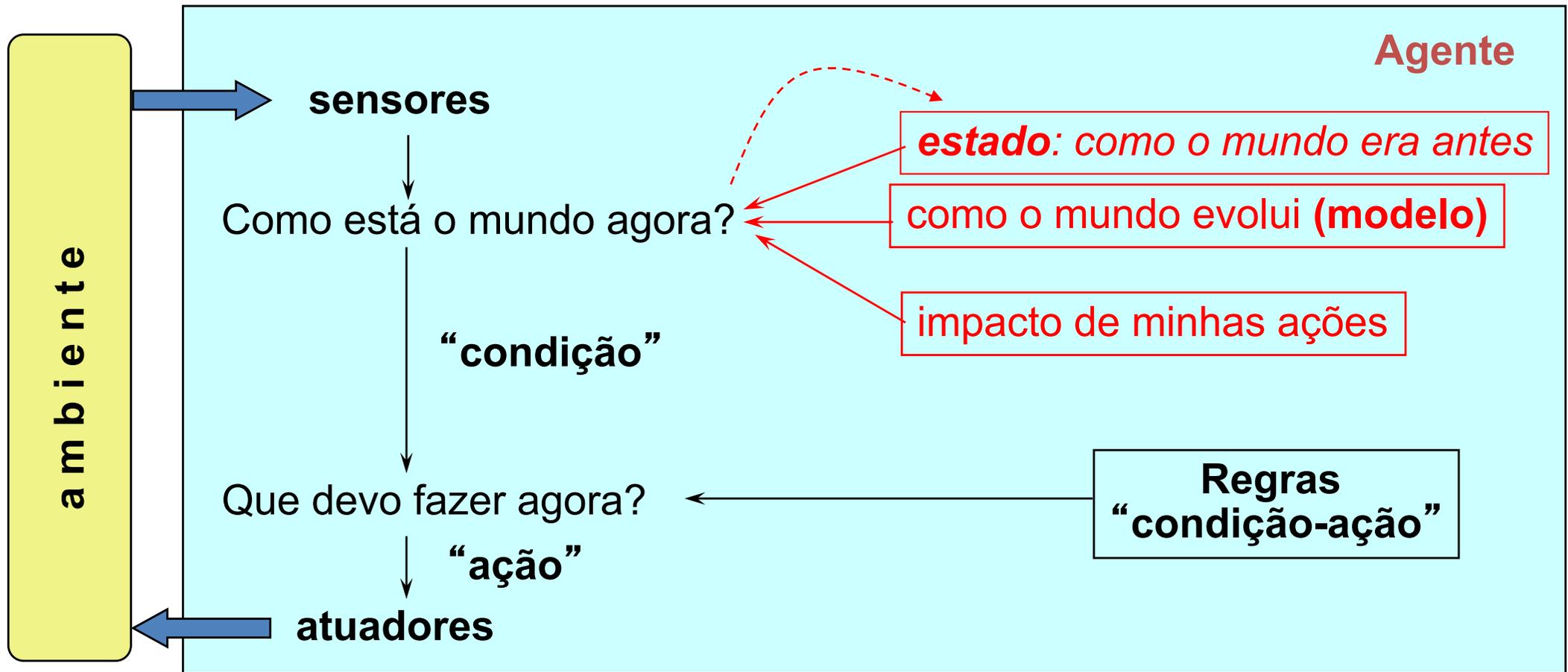
- Um agente puramente reativo que tenha observação parcial do ambiente pode cair em *deadlocks* ou *loops* infinitos.
- A forma mais efetiva de lidar com observabilidade parcial é “lembrar e imaginar a parte do mundo que não está observável no momento”, i.e, manter um estado interno que dependa da história perceptual passada e reflita (ao menos em parte) aspectos não observados no estado atual.

Necessidade de um modelo (2)

- Neste novo agente, para determinar como o mundo está num determinado momento, ele usa:
 - informações **perceptuais atuais** (como o agente reativo)
 - seu **estado** interno
 - informações a respeito de **como o mundo evolui**, independentemente de suas ações (**modelo do mundo**)
 - informações a respeito do **impacto/efeito de suas próprias ações** no mundo

e, com isso, atualiza seu estado interno.

Agente baseado em modelo



- Desvantagem: pouca autonomia
 - não tem objetivo, não encadeia regras
- Ambientes: determinístico e pequeno
 - pode lidar com a observabilidade parcial

Ex.: carro mudando de faixa, no trânsito (percepção parcial, precisa do modelo do mundo pra prever onde estão os outros carros).

function Agente-Com-Estado (*percept*) **return** uma ação

static: *estado* – uma descrição do estado corrente do mundo

regras – um conjunto de regras condição-ação

ação – a ação mais recente, inicialmente nula

estado ← **Atualiza-Estado**(*estado*, *ação*, *percept*)

regra ← **Acha-Regra**(*estado*, *regras*)

ação ← **Regra-Ação** [*regra*]

return *ação*

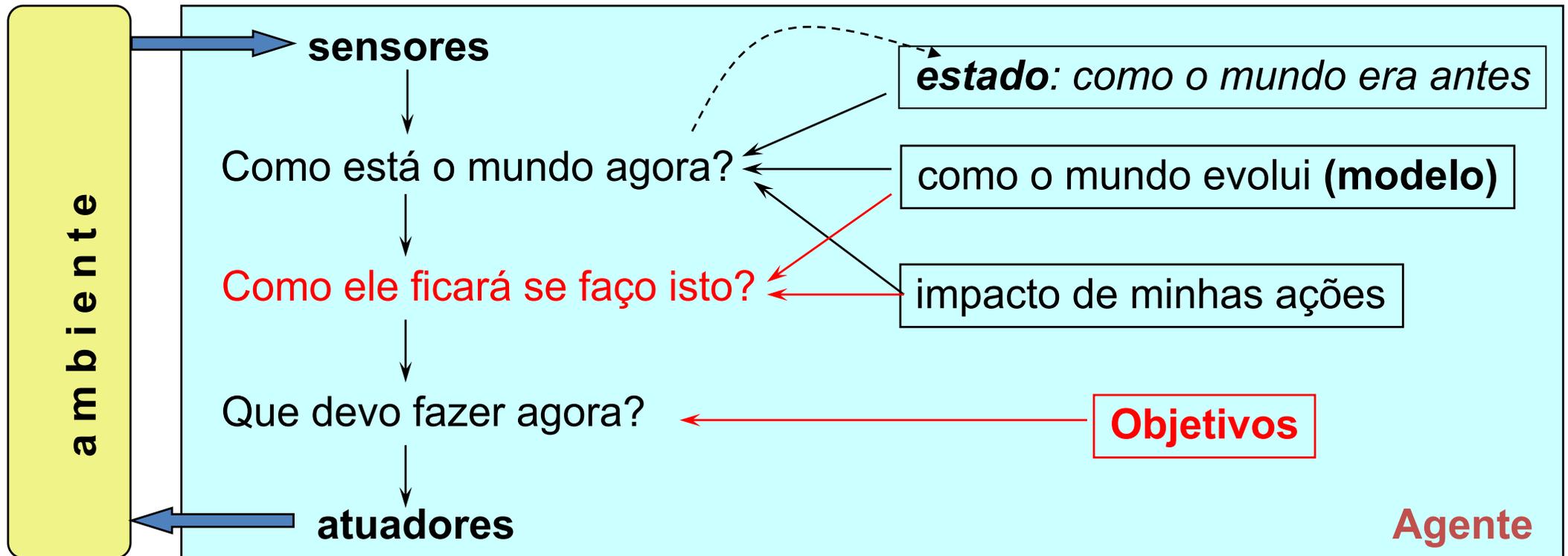
Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- **Agente baseado em objetivos**
- Agente baseado em utilidade
- Agente aprendiz

Necessidade de metas/objetivos

- Além do estado interno, um agente precisa de alguma informação a respeito de metas, indicando situações desejáveis, para decidir a melhor ação a executar.
- Assim, pode combinar as informações do impacto de suas ações com seus objetivos, de modo a fazer considerações acerca do futuro (predições) e decidir melhor suas ações.
 - O agente poderá ter que considerar longas seqüências de ações encadeadas para poder atingir sua meta → **busca** e **planejamento** são subáreas de IA que visam determinar a **seqüência de ações** que leva o agente ao objetivo.

Agente baseado em objetivo



- Vantagens e desvantagens:
 - Mais *complicado e ineficiente*, porém mais *flexível, autônomo*
 - Não trata objetivos conflitantes
- Ambientes: determinístico

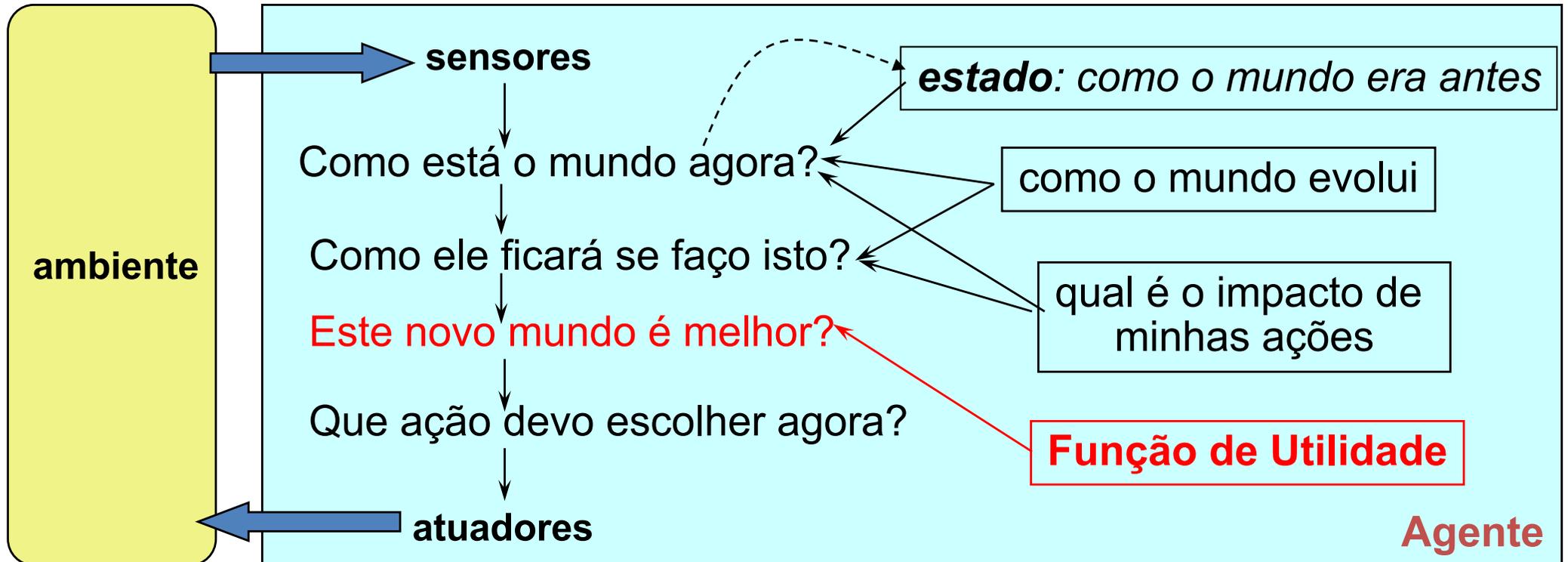
Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- **Agente baseado em utilidade**
- Agente aprendiz

Medida de desempenho mais geral: função de utilidade

- Uma função de utilidade mapeia um estado (ou seqüência de estados) em um número real que descreve o **grau de satisfação** associado ao estado.
- Permite decisões racionais quando há:
 - Objetivos conflitantes (ex: velocidade e segurança) → neste caso a função de utilidade define um compromisso adequado entre eles.
 - Múltiplos objetivos, nenhum dos quais se tem certeza de ser atingido → a função de utilidade permite um balanceamento entre a possibilidade de sucesso com a importância de cada objetivo.

Agente baseado em utilidade

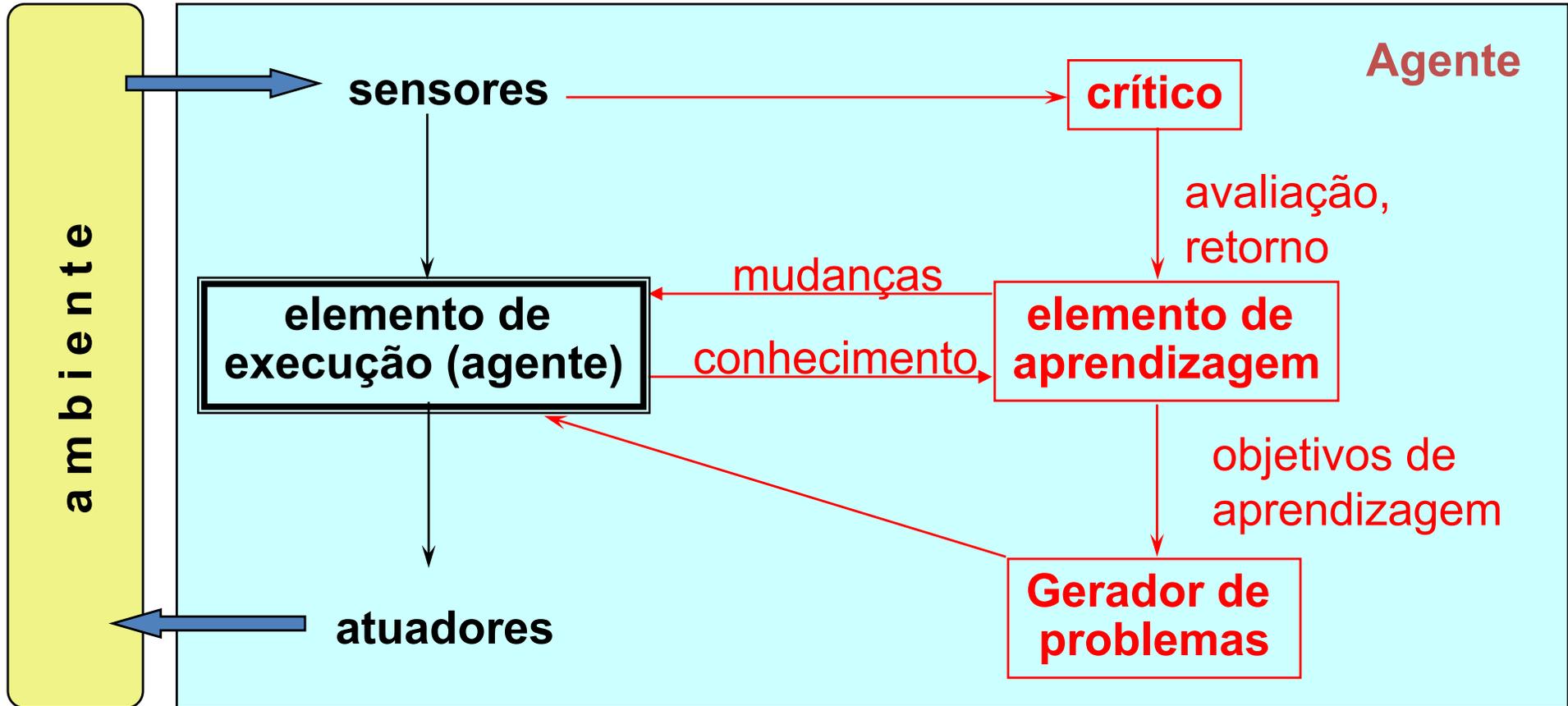


- Ambiente: sem restrição
- Desvantagem: não tem adaptabilidade (não aprende)

Arquiteturas

- Agente tabela
- Agente reativo
- Agente baseado em modelo
- Agente baseado em objetivos
- Agente baseado em utilidade
- **Agente aprendiz**

Agente que aprende



- Ambiente: sem restrição
- Vantagem: tem adaptabilidade (aprende)

Evolução da noção de agente além das fronteiras da IA....

- Agentes em IA
 - Metodologia (metáfora) para projeto de sistemas
 - Sistemas multiagentes e robótica
- Agentes em Computação
 - **Adoção de uma nova metáfora** (antropomórfica e sociológica). Extrapolação de OOP
 - Integração de técnicas de IA
 - Novas tecnologias próprias à Web (ex. mobilidade)
 - Marketing (moda)
- Agentes: técnica ou metodologia ?

Desenvolver agentes inteligentes

- Projeto:
 - Modelar tarefa em termos de ambiente, percepções, ações, objetivos e utilidade
 - Identificar o tipo de ambiente
 - Identificar a arquitetura de agente adequada ao ambiente e tarefa
- Implementação
 - O gerador e o simulador de ambientes
 - Componentes do agente (vários tipos de conhecimento)
 - Testar o desempenho com diferentes instâncias do ambiente

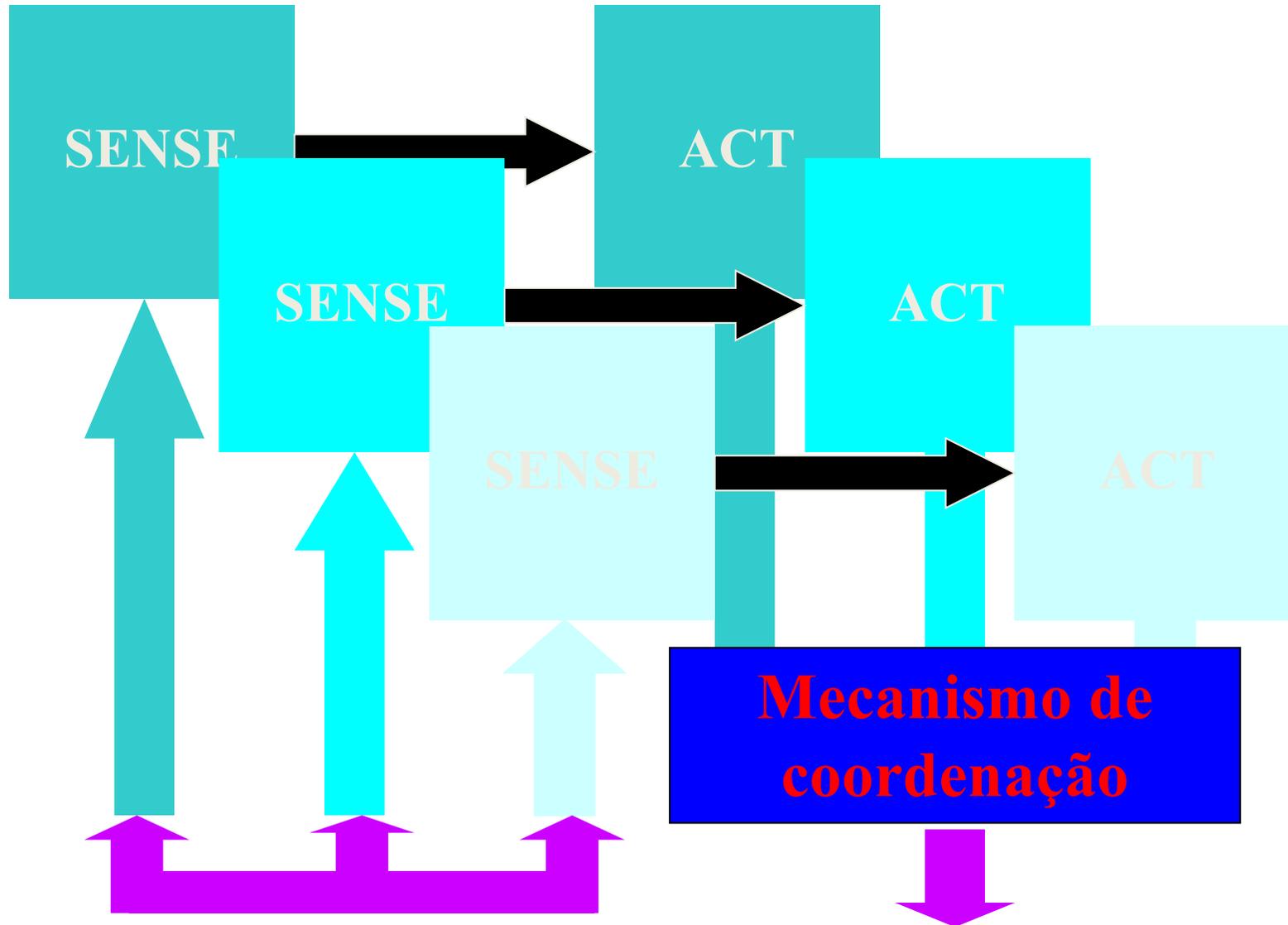
EXEMPLO DE AGENTE REATIVO

ROBÓTICA MÓVEL INTELIGENTE

Arquiteturas Reativas para Robôs

- Surgidas no final dos anos 80.
- Fundamentadas em estudos do comportamento animal (Etologia) → baseadas em comportamentos.
- Baseadas em processamento **paralelo** (vários comportamentos simultaneamente ativos).

Arquiteturas Reativas



Mecanismo de coordenação

- **Coordenação Competitiva:** a ação resultante num dado instante é selecionada a partir de uma competição entre os comportamentos ativos (um vence).
- **Coordenação Cooperativa:** a função de coordenação produz uma ação resultante para a qual contribuem todos os comportamentos ativos.

Estudo de Caso: REACT

Comportamentos Reativos para Robôs Móveis



LTI
Laboratório de
Técnicas Inteligentes

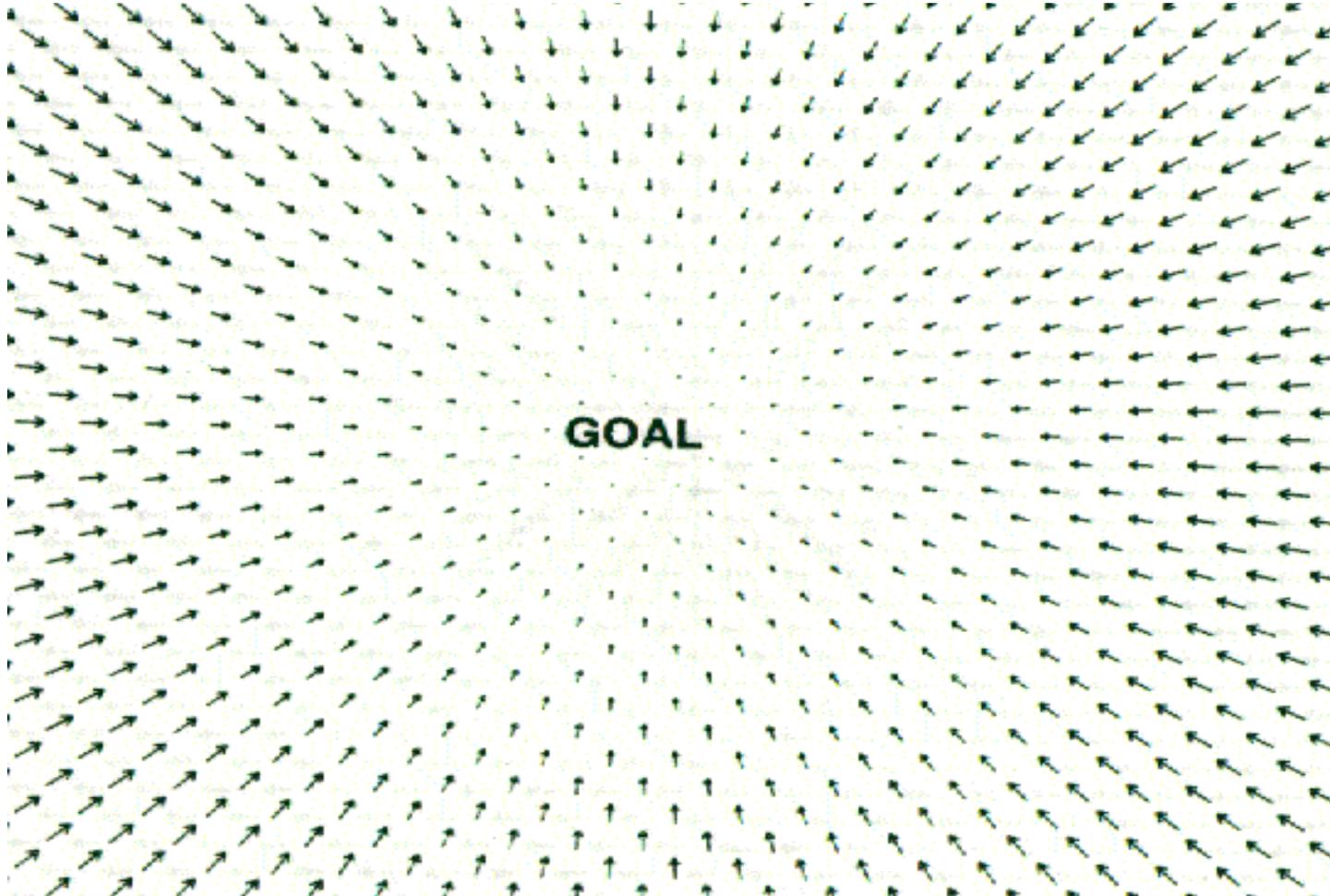
REACT

- Arquitetura baseada em *Motor Schemas*
- *Motor Schema* \equiv comportamento
 - Comportamentos são divididos em:
 - módulo de percepção
 - módulo de codificação
- Saída do *Motor Schema* \rightarrow *Vetor*
 - representa a ação a ser executada
 - Magnitude \equiv velocidade
 - direção \equiv rotação

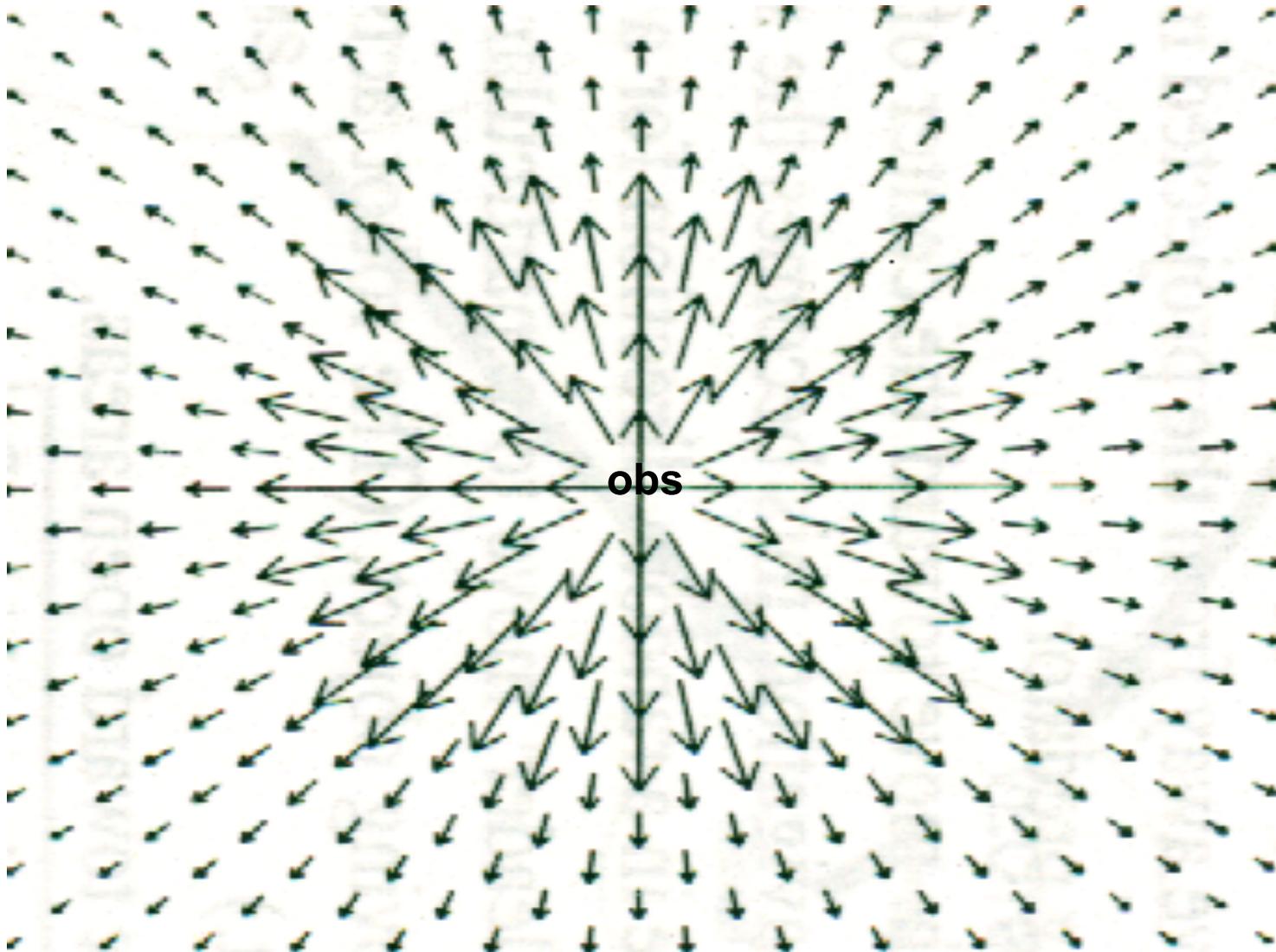
REACT

- Módulo de codificação:
 - Mapeamento contínuo: percepções → ações
 - Usa o Método de Campos Potenciais
- Coordenação dos comportamentos:
 - Abordagem cooperativa → Superposição dos campos de força
- Alguns comportamentos na REACT:
 - avoidCollision
 - moveToGoal

REACT – moveToGoal

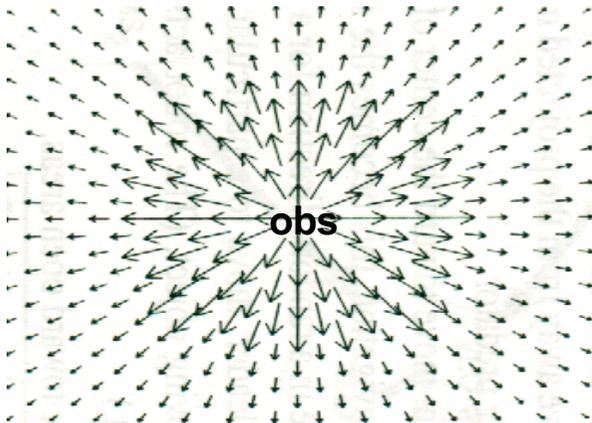
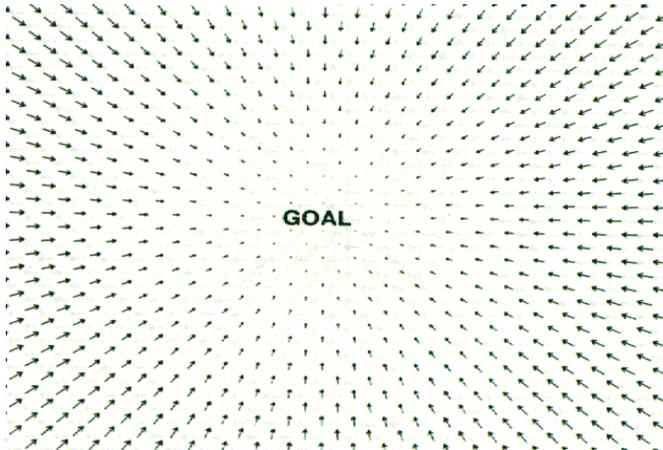


REACT - avoidCollision

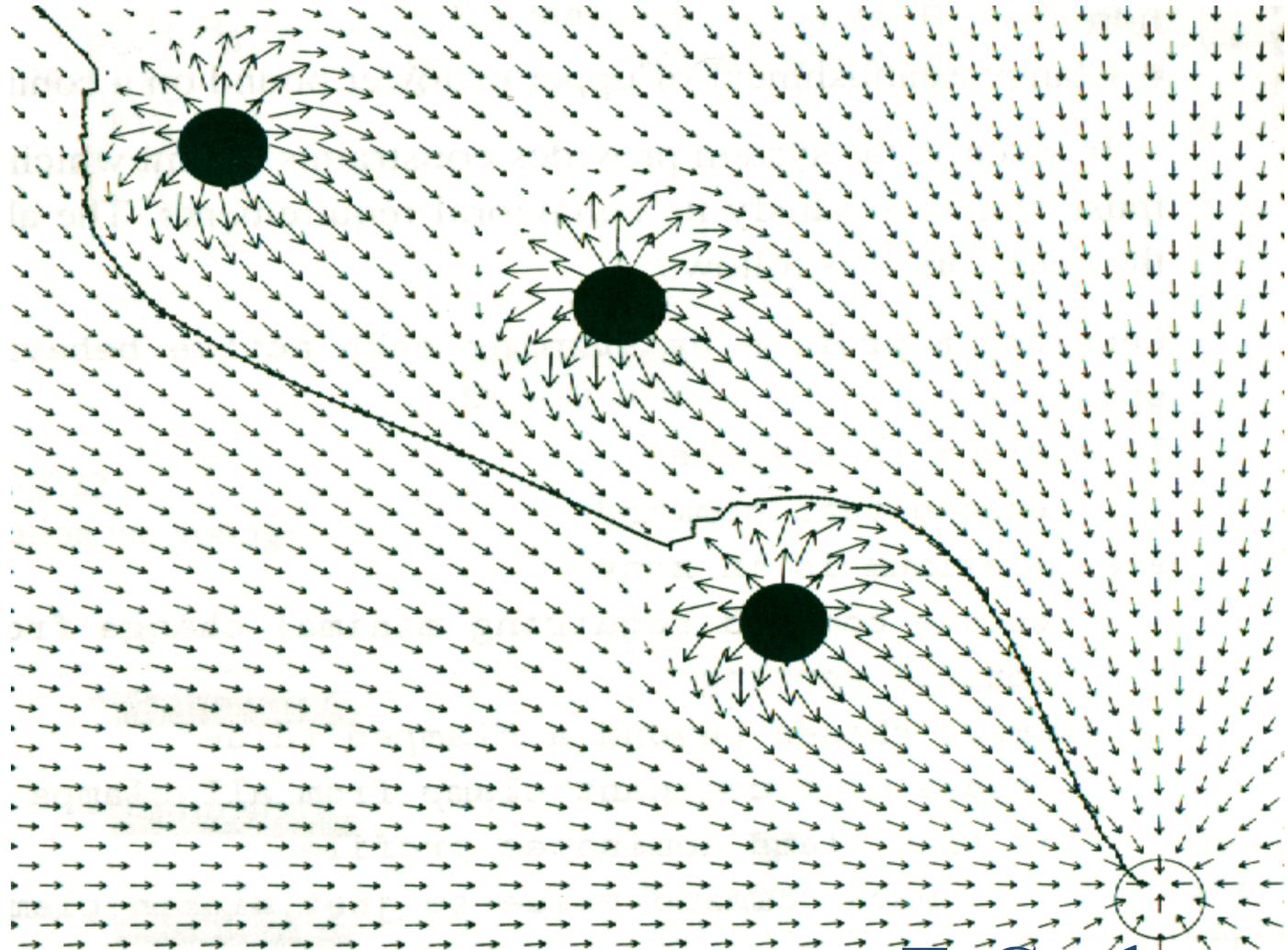


REACT

moveToGoal

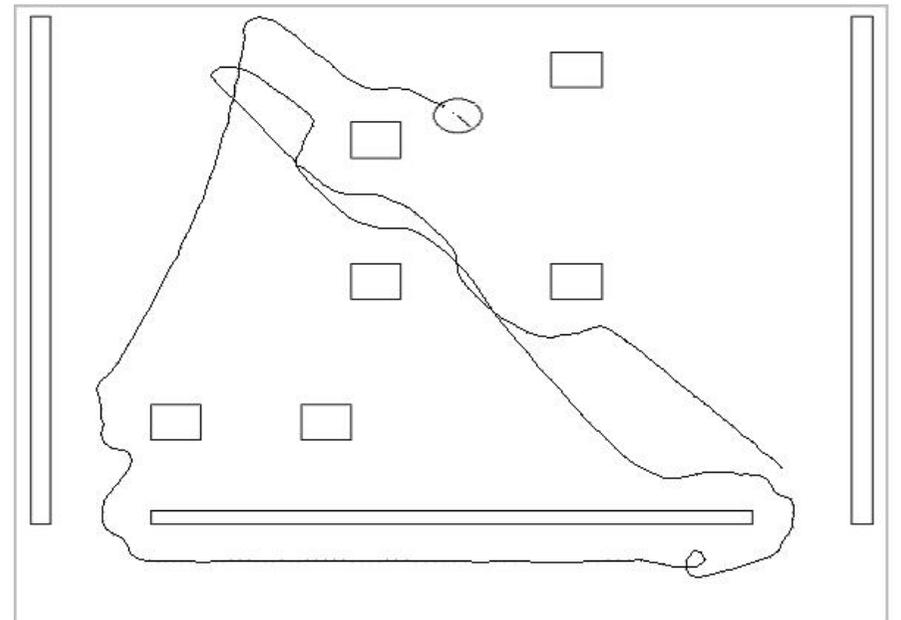
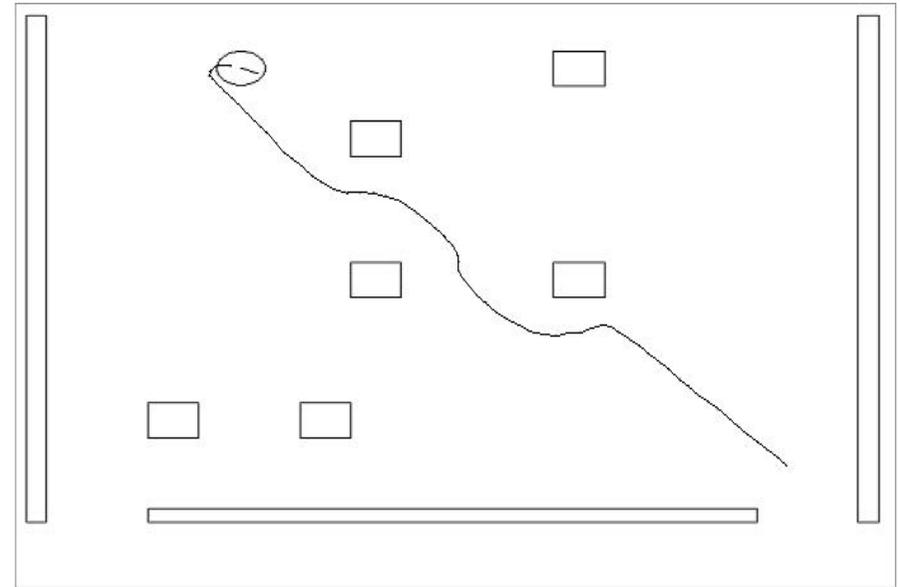
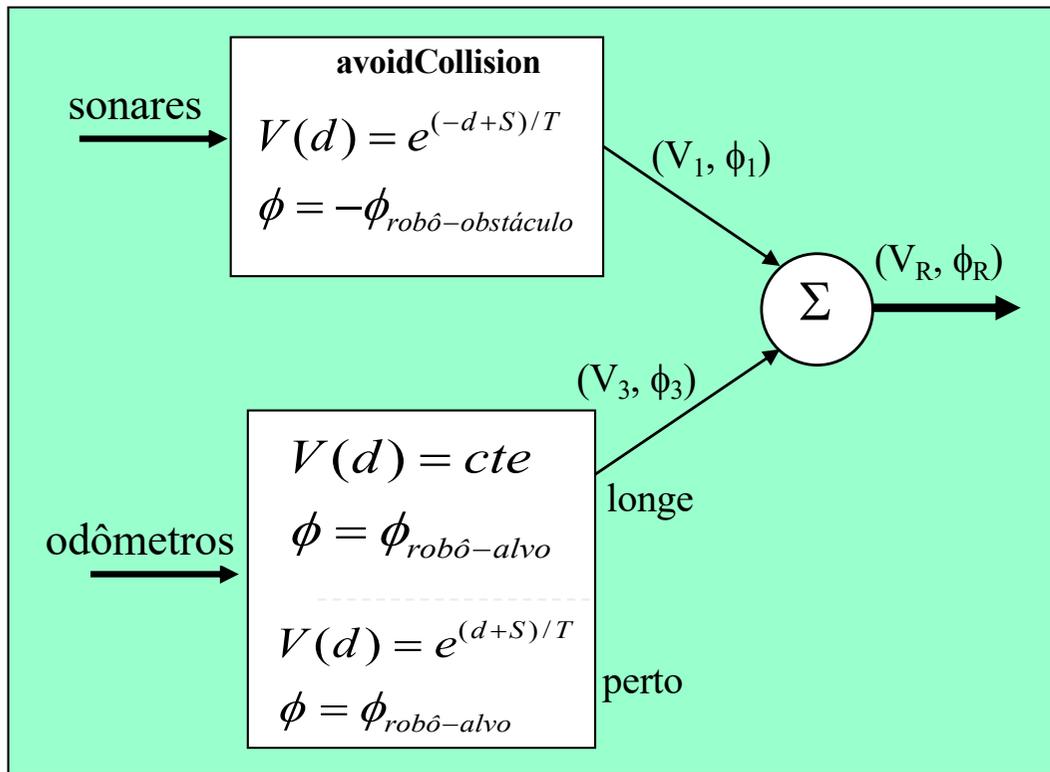


avoidCollision



avoidCollision + moveToGoal

REACT



Arquiteturas Reativas: Vantagens

- Comportamentos reativos são normalmente simples de projetar.
- Paralelismo, simplicidade de comportamentos individuais e ligação direta *SENSE-ACT* permitem operação em **tempo real**.
- Processamento **local** da informação sensorial.
- Prototipação rápida para poucos comportamentos.

Arquiteturas Reativas: Problemas

- Implementação de um grande conjunto de comportamentos é uma tarefa difícil.
- Combinação de comportamentos reativos não garante sucesso na execução da tarefa (defensores da abordagem falam em *inteligência emergente*).
 - Ex: na REACT, pode haver pontos de campo nulo.
- Difícil definição de um conjunto mínimo de comportamentos reativos no caso geral.