

## Agentes Inteligentes

### O que é um agente?

Um agente é algo que percebe seu ambiente através de sensores e atua no ambiente através de atuadores (*effectors*).

- Agente Humano:
  - Sensores: Olhos, ouvidos, nariz, tacto, paladar.
  - Atuadores: Pernas, braços, mãos.
- Agente Robótico: câmeras, sonares, sensores de infra-vermelhos, motores, rodas, etc.
- Agente em Software: possui *strings* de bits codificadas como percepção e ação.

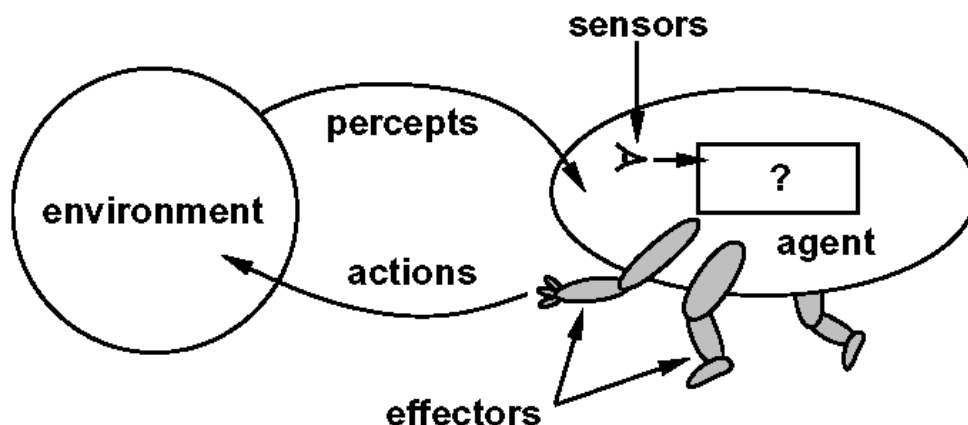


Figura -1:Agentes interagem com o ambiente através de sensores e atuadores

### Ação de um agente

```
Environment e;
RuleSet r;
while (true) {
    state = senseEnvironment(e);
    a = chooseAction(state, r);
    e.applyAction(a);
}
```

### O que é um Agente Racional?

- O objetivo da IA, segundo *Russel & Norvig* é projetar agentes que façam um bom trabalho agindo no seu ambiente. O princípio básico da utilização de agentes é que eles devem "saber das coisas" (*knowthings*).
- Um Agente Racional Ideal é aquele que para cada possível seqüência de percepção, realiza uma ação que maximiza seu desempenho (mapeamento ideal), tendo como base as evidências fornecidas pela seqüência de percepção e pelos conhecimentos previamente existentes no agente.
- Especificar que ações um agente deve tomar em resposta a qualquer seqüência de percepções, leva ao projeto de um agente ideal.
- Mapeamento entre percepções e ações.
- Exemplo: agente aspirador.

## Mapeamento ideal: da seqüência de percepção até a ação

- O comportamento de um agente depende de sua seqüência de percepção a sua conclusão (ação);
- Pode-se descrever qualquer agente através de uma "tabela" de respostas (ações) perante as suas percepções;
- Esta tabela representa o mapeamento entre a seqüência de percepção à ação.
- Exemplo: a função raiz quadrada de uma calculadora;

<pre>function SQR(x)      z ← 1.0      repeat until  z<sup>2</sup> - x      &lt; 10<sup>-15</sup>          z ← z-(z<sup>2</sup>-x)/(2z)      end      return z</pre>		<pre>#include #include #include  float rq(float x) { float z=1; do { z = z-((pow(z,2)-x)/(2*z)); } while (pow(z,2) &lt; 10e-15); return(z); }</pre>
Percept x	Action z	<pre>void main (void) { clrscr(); float _x; for (_x = 1; _x &lt; 2; _x=_x+0.1) printf("%1.15f - %1.15f\n", _x,rq(_x)); float n=10e-15; printf("\n%1.15f",n); getch(); }</pre>
1.0	1.0000000000000000	
1.1	1.049999952316284	
1.2	1.100000023841858	
1.3	1.150000095367432	
1.4	1.200000047683716	
1.5	1.2500000000000000	
1.6	1.300000071525574	
1.7	1.350000143051147	
1.8	1.400000095367432	
1.9	1.450000047683716	
Tabela -1: Parte de um mapeamento ideal da raiz quadrada, e seu correspondente programa em C.		

## O que é um Agente Autônomo?

- Agentes Autônomos são sistemas computacionais que operam em ambientes dinâmicos e imprevisíveis.
- Eles interpretam dados obtidos pelos sensores que refletem eventos ocorridos no ambiente e executam comandos em "motores" que produzem efeitos no ambiente.
- O grau de "autonomia" de um agente está relacionado à capacidade de decidir por si só como relacionar os dados dos sensores com os comandos aos motores em seus esforços para atingir seus objetivos satisfazer motivações, etc.

## Construção de agentes

- Estruturas de dados internas são atualizadas via percepções e usadas para tomar a decisão das ações a executar (melhor ação).

```

function SKELETON-AGENT(percept) returns action
  static: memory, the agent's memory of the world

  memory ← UPDATE-MEMORY(memory, percept)
  action ← CHOOSE-BEST-ACTION(memory)
  memory ← UPDATE-MEMORY(memory, action)
  return action

```

Figura -1: Estrutura de um agente

## Tipos de Agentes

- **Software Agents (softbots)**
  - Agentes são considerados entidades computacionais baseados na idéia de que os usuários necessitam apenas especificar um objetivo em alto nível, ao invés de utilizar instruções explícitas, deixando as questões de como e quando agir a cargo do agente.
  - Aplicações: interfaces amigáveis, cartografia, auxílio ao ensino, auxílio ao diagnóstico médico.
- **Hardware Agents (robôs, *embedded systems*)**
  - Agentes que operam em ambientes físicos (apesar de que muitas vezes estes ambientes são simulados).
  - Terceira geração de robôs, capazes de detectar mudanças ambientais e, através da reavaliação de seus objetivos, encontrar uma nova seqüência de ações capazes de persegui-los, sem que esta seqüência tivesse sido prevista.

Tipo de Agente	Percepção	Ação	Objetivos	Ambiente
Sistema de Diagnóstico Médico	Sintomas, exames, respostas do paciente.	Perguntas, testes, tratamentos.	Curar o paciente, reduzir custos.	Paciente hospital
Sistema de Análise de Imagens de Satélite	<i>Pixels de intensidade variada e cores</i>	Fornecer uma classificação para a imagem	Classificar corretamente	Imagens de Satélites Orbitais
Robô manipulador	<i>Pixels de intensidade variada</i>	Pegar diferentes tipos de peças e colocá-las em <i>containers</i> .	Colocar as peças nos <i>containers</i> corretos	Linha de montagem com peças
Controlador de Refinaria	Leituras de Temperatura e Pressão	Abrir e fechar válvulas, controlar a temperatura.	Maximizar a pureza, custos, segurança.	Refinaria
Tutor Interativo de uma língua	Palavras digitadas	Propor exercícios, sugestões, correções.	Maximizar o resultado do aluno	Conjunto de Estudantes

Tabela -1:Exemplos de tipos de agentes e suas descrições PAGE

## Complexidade dos Agentes

- **Agentes Reflexivos**
  - Não tem memória.
  - Quando cessa a percepção, cessa a ação.
  - Baseados em tabelas de regras condição-ação (regras *if-then*)

- *If car in front is braking (brake-light on), then initiate breaking.*

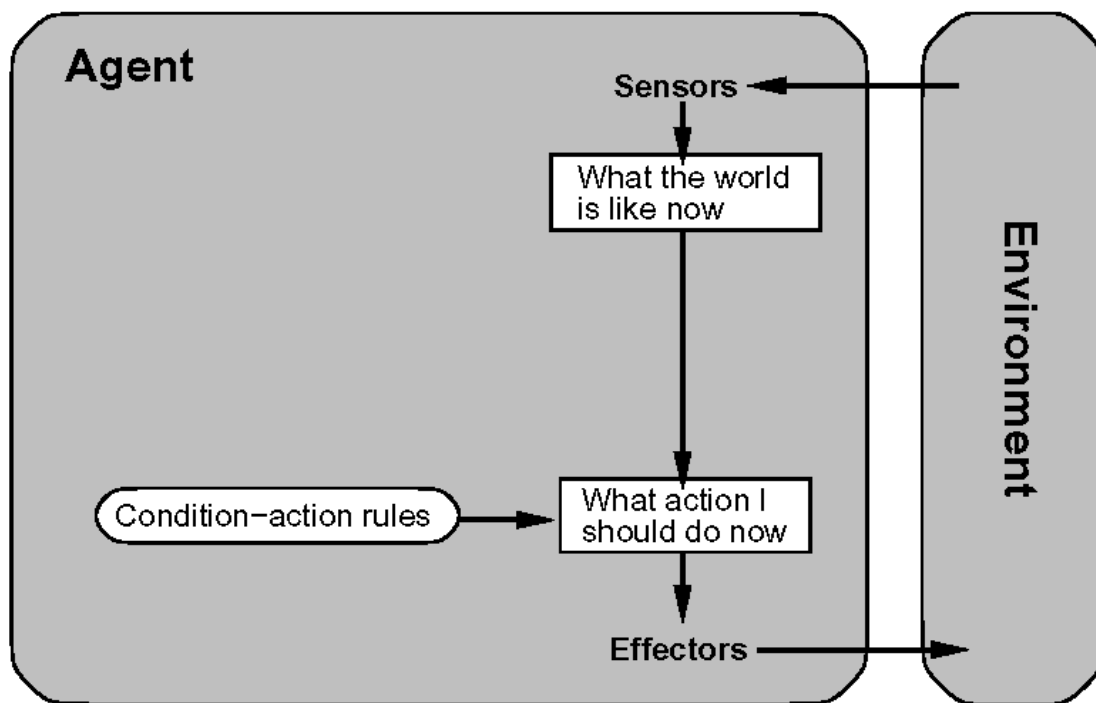


Figura -1: Diagrama esquemático de um agente reflexivo simples

```

function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules, a set of condition-action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action

```

Figura -2: Estrutura do agente reflexivo, o qual trabalha em busca de uma regra que combine com a situação corrente e então efetua a ação correspondente a aquela regra.

### ◦ Agentes com Estados Internos

- Mantêm um estado interno (representação do mundo)
- Guarda informações que não são percebidas no momento
- Como o mundo evolui
- O que as ações provocam no mundo

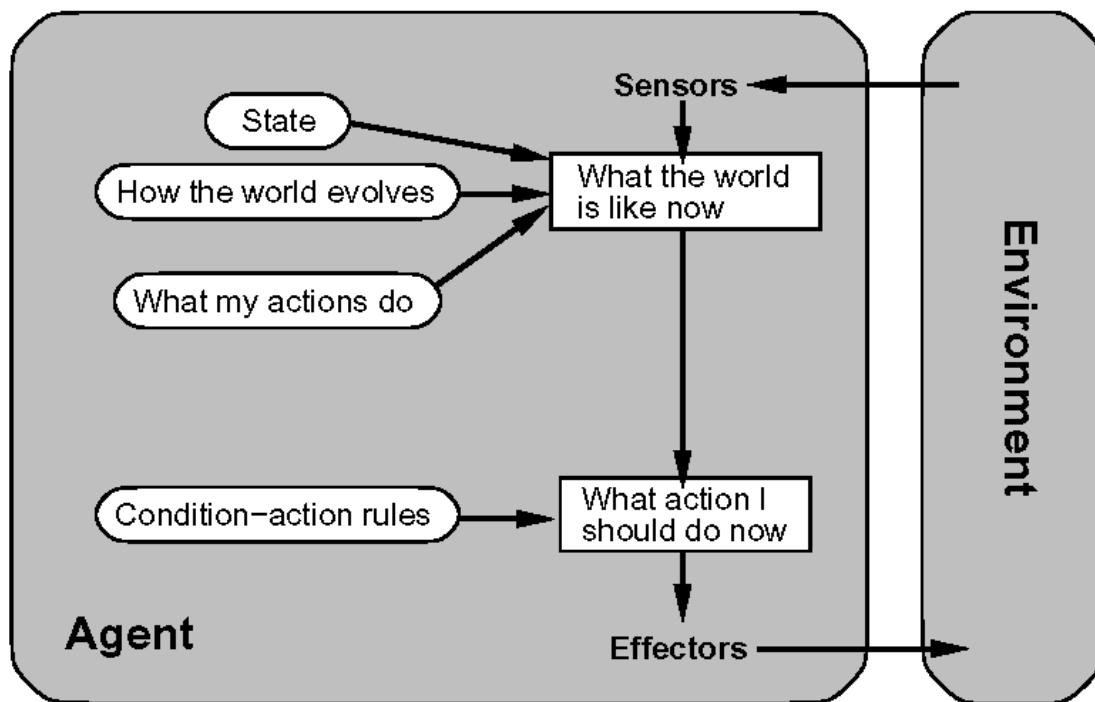


Figura -1: Agente reflexivo com estado interno

```

function REFLEX-AGENT-WITH-STATE(percept) returns action
  static: state, a description of the current world state
           rules, a set of condition-action rules

  state ← UPDATE-STATE(state, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  state ← UPDATE-STATE(state, action)
  return action
  
```

Figura -2: Estrutura do agente com estados internos, o qual trabalha em busca de uma regra que combine com a situação corrente (definida pela percepção e o estado interno armazenado) e então efetua a ação correspondente a aquela regra.

### ◦ Agentes com Metas

- Descrição do estado do mundo e do objetivo a atingir
- Exemplo: Chegar a Florianópolis
- Resolução de problemas por Pesquisa, Planejamento.
- Busca e Planejamento são subcampos da IA cujo objetivo é achar seqüências de ações que conduzam ao objetivo do agente.

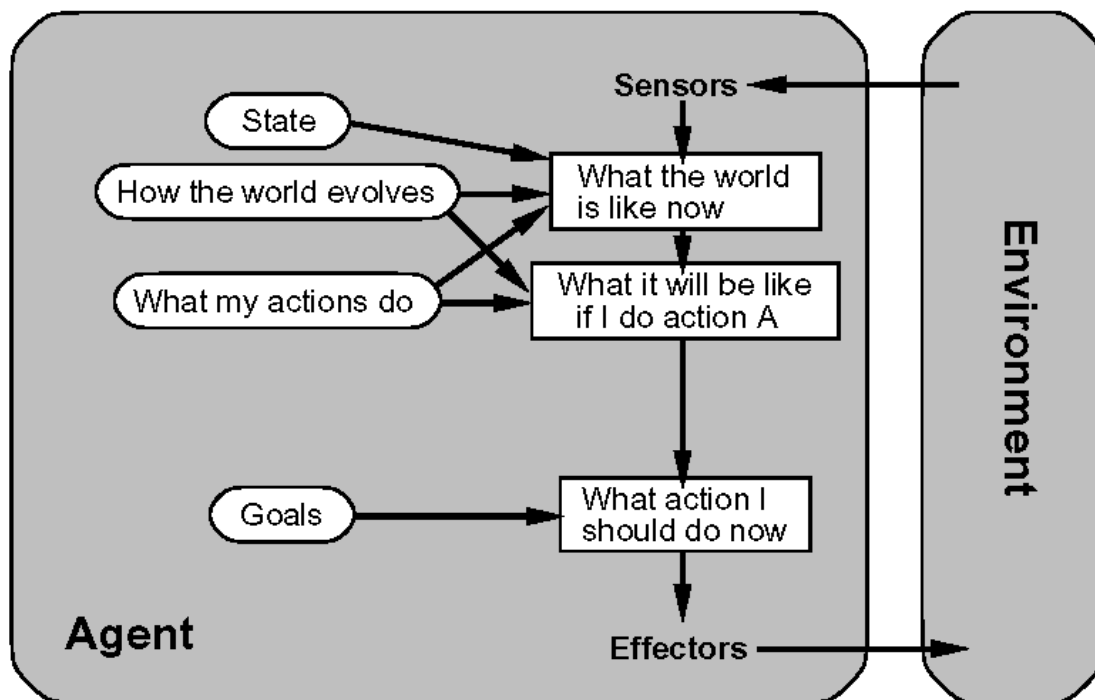


Figura -1: Agente com metas

### ◦ Agentes Baseados em Utilidade

- Utilidade é uma função que mapeia um estado em um número real que descreve o grau de "felicidade" associado ao estado.
- Permite decisões racionais em casos em que o objetivo tem algum "problema";
  - Quando existem objetivos conflitantes, tal como velocidade e segurança;
  - Quando existem vários objetivos, a utilidade "diz" qual tentar alcançar primeiro.

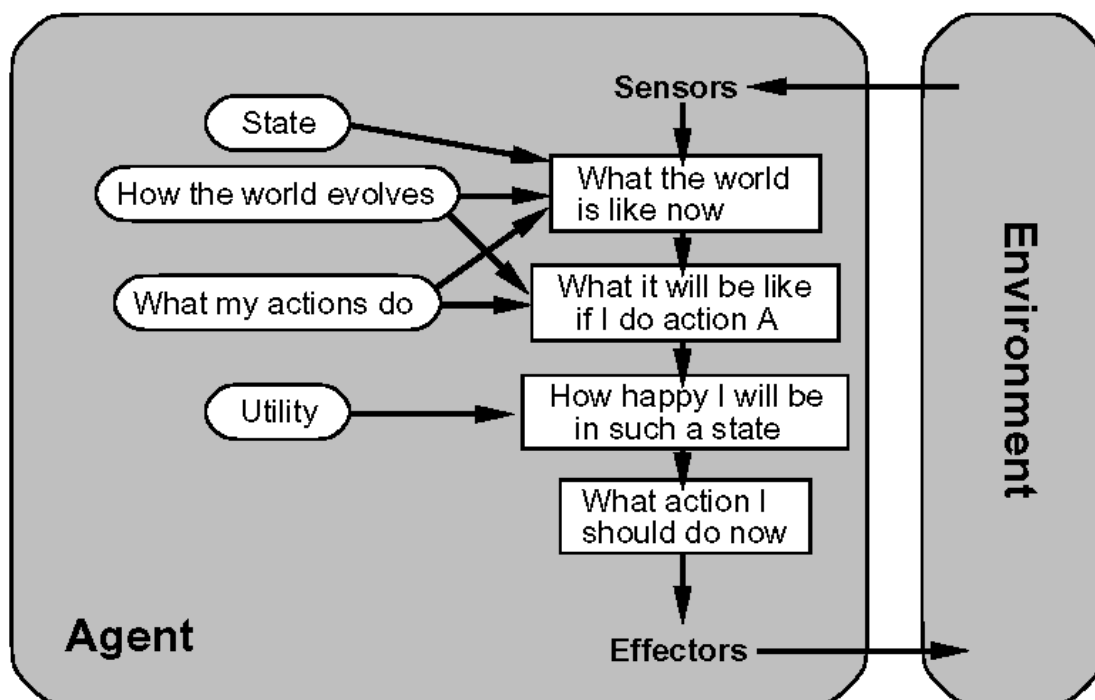


Figura -1: Agentes Baseados em Utilidade

## Ambientes

### ◦ Propriedades dos Ambientes

Acessível x Inacessível

Se o aparato sensor fornece acesso a uma descrição completa do

	ambiente.
Determinístico x Não-Determinístico	Se o próximo estado do ambiente pode ser completamente determinado pelo estado atual do ambiente e pelas ações selecionadas pelo agente.
Episódico x Não-Episódico	A experiência do agente é dividida em episódios. Cada episódio consiste na percepção do agente e na sua ação. Não existe passado nem futuro.
Estático x Dinâmico	Se o ambiente se altera enquanto o agente está pensando, então o ambiente é dinâmico para o agente.
Discreto x Contínuo	Se existe um número finito de diferentes percepções e ações possíveis, então o ambiente é discreto.

<b>Ambiente</b>	<b>Acessível</b>	<b>Determinístico</b>	<b>Episódico</b>	<b>Estático</b>	<b>Discreto</b>
Xadrez com Relógio	SIM	SIM	NÃO	SEMI	SIM
Xadrez sem Relógio	SIM	SIM	NÃO	SIM	SIM
<i>Poker</i>	NÃO	NÃO	NÃO	SIM	SIM
Gamão	SIM	NÃO	NÃO	SIM	SIM
Dirigir Táxi	NÃO	NÃO	NÃO	NÃO	NÃO
Diagnóstico Médico	NÃO	NÃO	NÃO	NÃO	NÃO
Análise de Imagens	SIM	SIM	SIM	SEMI	NÃO
Robô Manipulador	NÃO	NÃO	SIM	NÃO	NÃO
Controlador de Refinaria	NÃO	NÃO	NÃO	NÃO	NÃO
Tutor Interativo de Linguagens.	NÃO	NÃO	NÃO	NÃO	SIM

Tabela -1: Exemplos de ambientes e suas características

[\[Voltar\]](#)