

Introdução

Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo



O que é “inteligência artificial”?

Para responder a esta questão, antes temos de definir o que é “inteligência” e o que é “artificial”. A definição de “artificial” é simples; porém, quanto à definição de “inteligência”, há controvérsia.

Artificial (*Michaelis*)

é algo produzido pelo homem e não por causas naturais

Inteligência (*Michaelis*)

é a faculdade de aprender, compreender e adaptar-se

Outras acepções para o termo:

- **(teológica)** é um dom divino que nos torna semelhantes ao Criador
- **(filosófica)** é um princípio abstrato que é fonte da intelectualidade
- **(psicológica)** é a capacidade de resolver problemas com rapidez e êxito



Inteligência Artificial (IA)

Como não há consenso sobre o significado de “inteligência”, preferimos definir “inteligência artificial” enquanto área do conhecimento.

Inteligência Artificial (IA)

é a área da Computação que estuda como simular **comportamento inteligente** usando métodos computacionais.

Mas o que é comportamento inteligente?

Um computador pode pensar?

Um computador tem livre arbítrio?

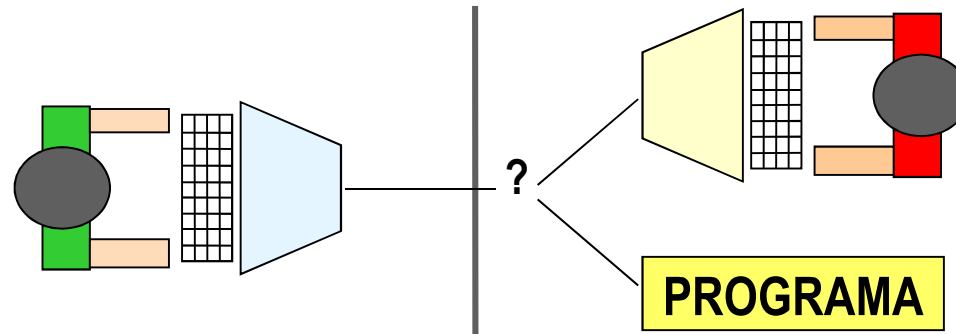
Um computador pode se comportar de forma inteligente?





Teste de Turing

Para evitar discussões filosóficas, Turing (1950) propôs um teste que consiste, basicamente, em entrevistar um agente num local remoto e decidir se este é uma pessoa ou um programa de computador.



Argumento de Turing

- Mesmo sem uma definição precisa de inteligência, podemos assumir que o ser humano é inteligente.
- Portanto, se um programa consegue se passar por um ser humano, podemos dizer que ele apresenta algum tipo de inteligência que, neste caso, só pode ser artificial.



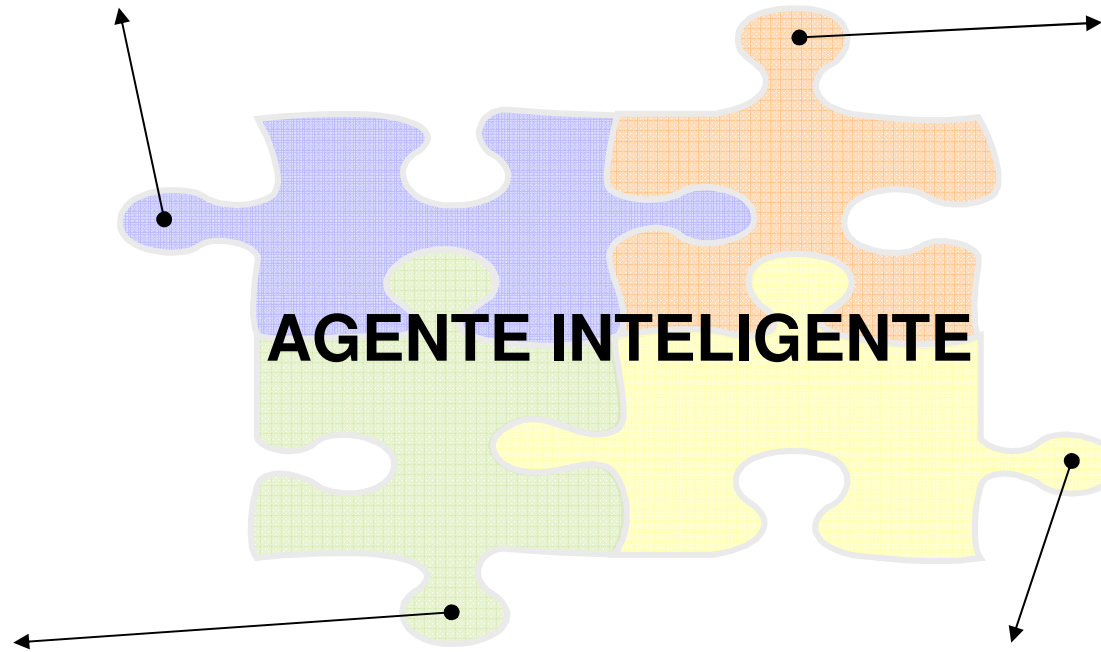
Capacidades exigidas pelo Teste de Turing

Processamento de Linguagem Natural

para que o agente possa se comunicar com o meio

Representação de Conhecimento

para que o agente possa guardar o que sabe



AGENTE INTELIGENTE

Raciocínio Automatizado

para que o agente possa usar o que sabe

Aprendizado de Máquina

para que o agente possa atualizar o que sabe



Breve histórico da IA

O início (1943-1950)

- modelo de neurônios artificiais (*McCulloch & Pitts*, 1943)

Fase de grande entusiasmo (1951-1969)

- jogar xadrez (*Shannon*, 1950; *Turing*, 1953)
- provar teoremas (*Newell & Simon*, 1956)
- planejar tarefas (*Green*, 1963)
- comunicação em linguagem natural (*Weizenbaum*, 1965)
- aprender por analogia (*Evans*, 1968)

Fase difícil (1970-1980)

- problemas de capacidade de processamento e armazenamento
- publicação da teoria da complexidade computacional (*Cook*, 1971)

O ressurgimento (1981-presente)

- projeto de computador japonês de 5ª geração (1980)
- pesquisas voltadas a aplicações práticas em áreas específicas

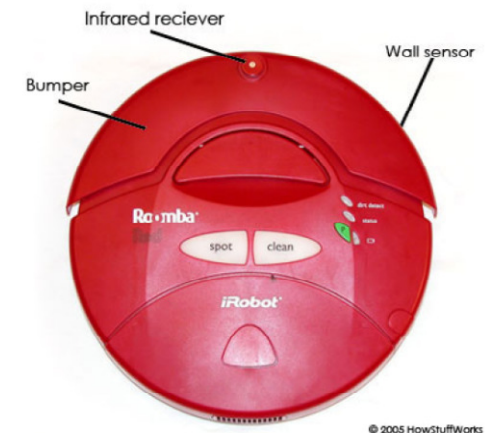


Áreas de aplicação de IA

Há diversas áreas de aplicação de inteligência artificial.

Exemplos

- Eletrodomésticos
- Jogos e brinquedos eletrônicos
- Robótica e automação industrial
- Verificação automática de software
- Otimização e controle de processos
- Processadores de linguagem natural
- Bancos de dados dedutivos e mineração de dados
- Aprendizagem, planejamento e escalonamento de tarefas
- Reconhecimento de imagens, sons, cheiros e sabores

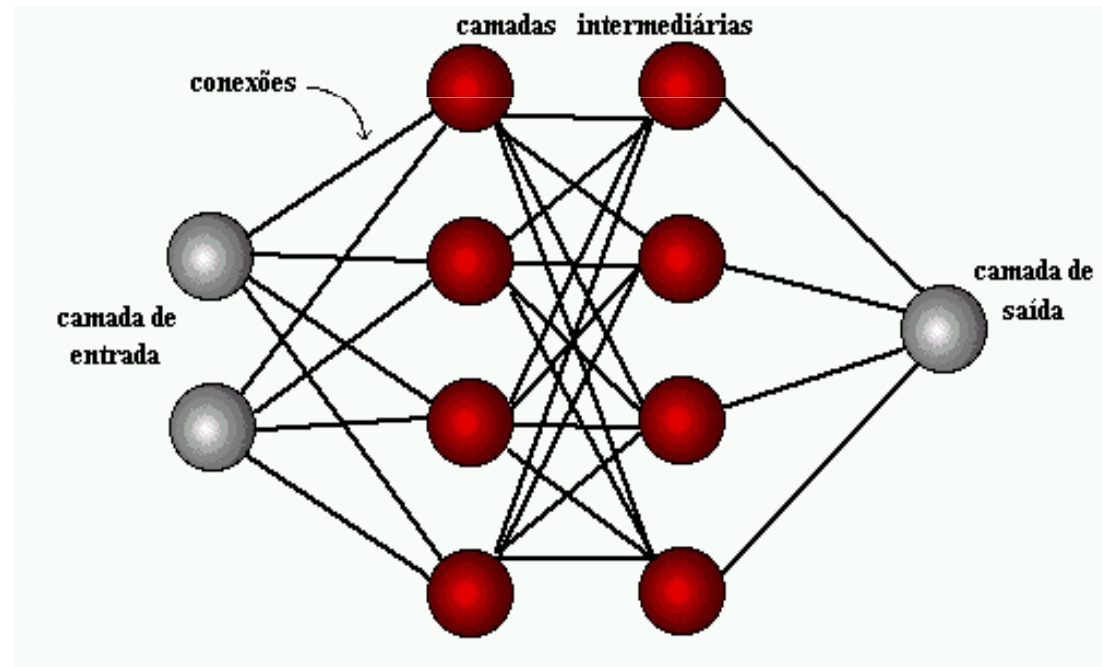
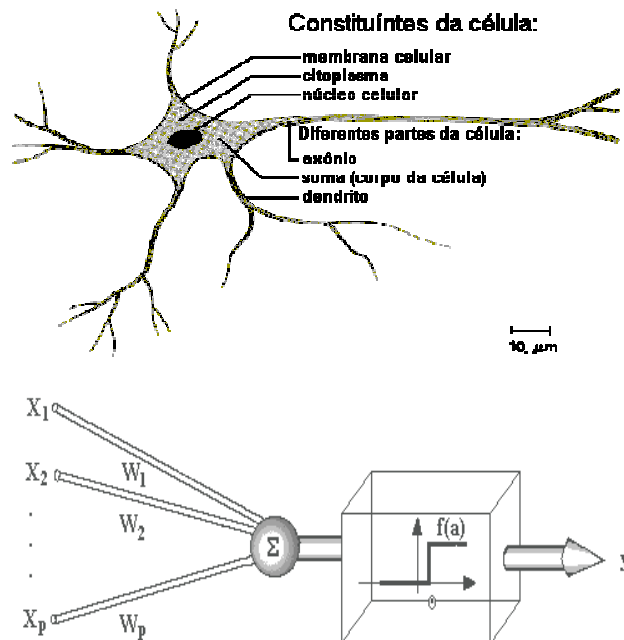




Principais abordagens em IA

Conexionista

- **Hipótese:** um modelo preciso do cérebro humano é suficiente para reproduzir inteligência
- **Aplicação:** problemas imprecisos definidos por exemplos (e.g., reconhecimento de voz)
- **Principal contribuição:** **redes neurais artificiais**



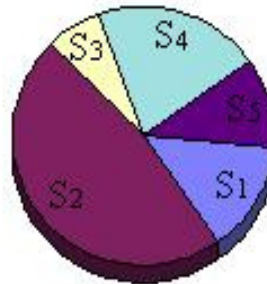


Principais abordagens em IA

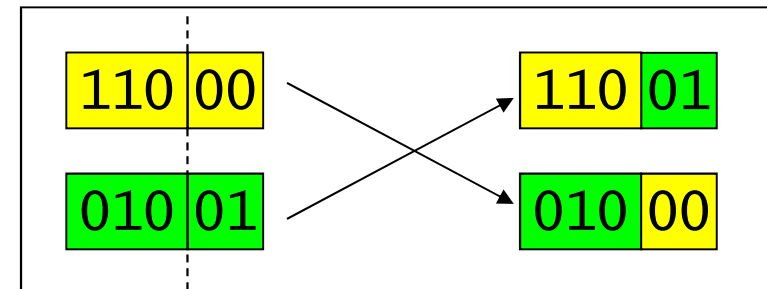
Evolucionista

- **Hipótese:** a solução de um problema pode ser obtida pela evolução de uma população de indivíduos que carregam genes com informações relevantes para o problema
- **Aplicação:** problemas de otimização difíceis (e.g., escalonamento)
- **Principal contribuição:** **algoritmos genéticos**

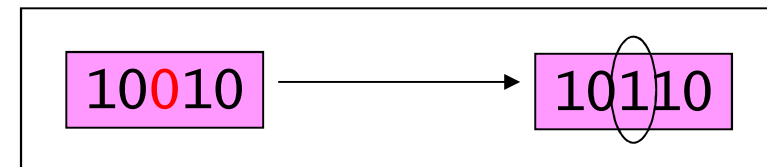
Indivíduo	Aptidão	Aptidão
S_i	$f(S_i)$	Relativa
S_1 10110	2.23	0.14
S_2 11000	7.27	0.47
S_3 11110	1.05	0.07
S_4 01001	3.35	0.21
S_5 00110	1.69	0.11



Cruzamento



Mutação





Principais abordagens em IA

Simbolista

- **Hipótese:** um conjunto de estruturas simbólicas e regras de manipulação destas estruturas são os meios necessários e suficientes para se criar inteligência
- **Aplicação:** problemas formalmente bem definidos (e.g., planejamento de tarefas)
- **Principal contribuição:** sistemas especialistas

Raciocínio indutivo

- **Conhece:** $p(a, b)$, $p(a, d)$, $p(d, e)$, $p(d, g)$, $p(e, f)$
- **Observa:** $a(a, e)$ e $a(d, f)$
- **Aprende:** $p(X, Y) \wedge p(Y, Z) \rightarrow a(X, Z)$

Raciocínio abdutivo

- **Conhece:** $p(a, b)$, $p(a, d)$, $p(d, e)$, $p(d, g)$, $p(e, f)$, $p(X, Y) \wedge p(Y, Z) \rightarrow a(X, Z)$
- **Observa:** $a(a, c)$
- **Explica:** $p(b, c)$ ou $p(d, c)$

Raciocínio dedutivo

- **Conhece:** $p(a, b)$, $p(a, d)$, $p(b, c)$, $p(d, e)$, $p(d, g)$, $p(e, f)$, $p(X, Y) \wedge p(Y, Z) \rightarrow a(X, Z)$
- **Conclui:** $a(a, c)$, $a(a, e)$, $a(a, g)$ e $a(d, f)$



O papel da lógica na Inteligência Artificial

A experiência mostra que...

um aspecto fundamental do comportamento inteligente é que ele é condicionado pelo **conhecimento** que um agente tem sobre seu mundo.

Mas, o que é conhecimento?

Quando ouvimos uma frase do tipo **“Ana sabe que ...”**, em geral, esperamos que ela seja completada com uma sentença como, por exemplo:

- **“está chovendo”**
- **“se está chovendo, então a rua está molhada”**

Isto sugere que, entre outras coisas, **conhecimento é uma relação entre um agente (“Ana”) e uma sentença declarativa (“está chovendo”).**



O papel da lógica na Inteligência Artificial

Outro aspecto fundamental do comportamento inteligente é que...

ele resulta de **raciocínio** correto sobre o conhecimento que se tem disponível.

Por exemplo, a partir do conhecimento representado pelas sentenças:

- *Está chovendo.*
- *Se está chovendo, então a rua está molhada.*
- *Se a rua está molhada, então a rua está escorregadia.*
- *Se a rua está escorregadia, devemos usar calçado que não escorrega.*
- *Se a rua não está escorregadia, então podemos usar qualquer calçado.*

Concluimos que a melhor coisa a fazer é:

- ***usar um calçado que não escorrega.***

**conhecimento
implícito!!!**



O papel da lógica na Inteligência Artificial

Assim, o principal papel da lógica na IA é...

garantir que novas informações possam ser corretamente extraídas do conhecimento explicitamente armazenado por um agente.

- Segundo [Shanahan, 1997]:
 - A melhor maneira de entender o comportamento inteligente é considerá-lo como resultado de um **raciocínio correto** sobre uma **representação correta**.
 - A **lógica simbólica** é o melhor formalismo para explicar as noções de representação correta e raciocínio correto.



Programação em lógica

- Embora existam outras abordagens computacionais interessantes para simulação de comportamento inteligente (*conexionista* e *evolucionista*), neste curso, adotaremos a abordagem *simbolista*.
- Mais precisamente, adotaremos a **programação em lógica** como paradigma para a construção de agentes inteligentes.

Programação em lógica é...

um formalismo lógico-computacional fundamentado em três princípios básicos:

- uso de **linguagem formal** para representação de conhecimento
- uso de **regras de inferência** para manipulação de conhecimento
- uso de uma **estratégia de busca** para controle de inferências



Programação em lógica: linguagem formal

- **Uma linguagem natural é ambígua**
 - *“Ana viu um homem numa montanha usando um binóculo”*
 - Quem usava o binóculo?
 - *“Ana, usando um binóculo, viu um homem numa montanha”*
 - *“Ana, estando numa montanha, viu um homem que usava um binóculo”*
- **Uma linguagem formal é precisa**
 - suas sentenças
 - são objetos (fórmulas) com significado único
 - têm sintaxe e semântica bem definidas
 - mas também pode ser menos expressiva



Programação em lógica: regra de inferência

- Regra de inferência é um padrão de manipulação sintática que:
 - permite criar novas fórmulas a partir de outras existentes
 - em geral, simulam formas de raciocínio válidas
- Exemplo (modus ponens):

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

- *Se neva, faz frio. Está nevando. Logo, está frio.*
- *Se vejo TV, fico com sono. Estou vendo TV. Logo, estou com sono.*

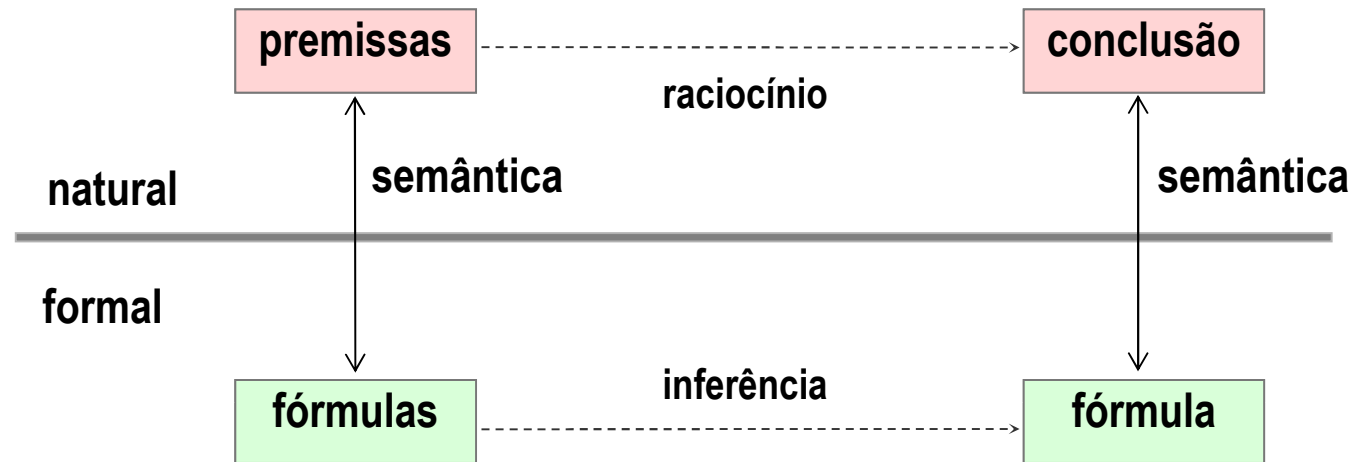


Programação em lógica: estratégia de busca

- Um agente pode ter uma enorme quantidade de conhecimento armazenado
- Assim como nós, ele precisa usar apenas parte de seu conhecimento para resolver um problema
- Estratégia de busca serve para decidir que parte do conhecimento armazenado deve ser explorada em busca da solução



Programação em lógica: idéia básica



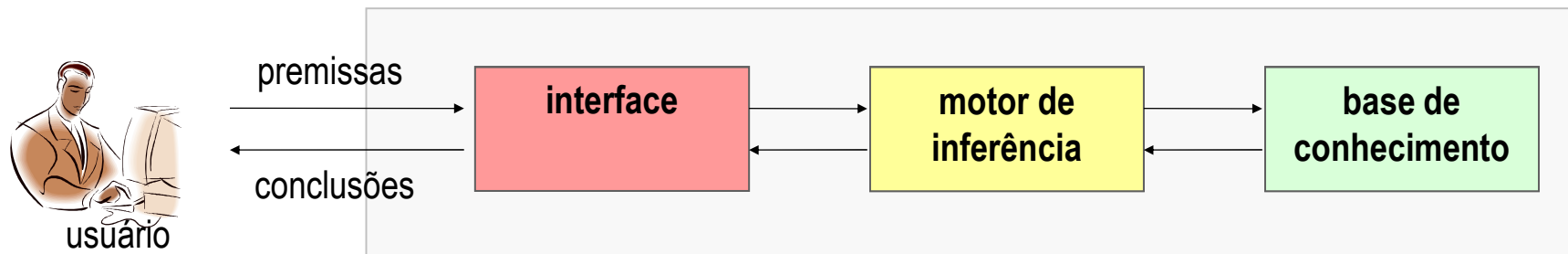
A idéia básica da programação em lógica é...

oferecer um arcabouço que permita inferir conclusões desejadas, a partir de premissas representando o conhecimento disponível, de uma forma que seja computacionalmente viável



O sistema Prolog

Prolog é o sistema de programação em lógica mais popular que existe!



- **Interface:** permite que o usuário entre com premissas codificadas em uma linguagem lógica e faça consultas para extrair conclusões destas premissas
- **Motor de inferência:** atualiza a base de conhecimento com premissas fornecidas pelo usuário e faz inferências para extrair informações implícitas
- **Base de conhecimento:** que armazena as premissas fornecidas pelo usuário



O sistema Prolog: vantagens

- Prolog permite representar o conhecimento que um agente tem sobre seu mundo de uma forma simples e direta, em uma linguagem de alto nível, tornando os programas mais compactos, flexíveis e inteligíveis.
- Prolog permite **programação declarativa**; em vez de especificar como o computador deve proceder para resolver um problema, precisamos apenas declarar o conhecimento que temos acerca do problema e, em seguida, consultar o sistema para que ele encontre a solução desejada.
- Em outras palavras, em Prolog, basta especificar corretamente o problema que o motor de inferência se encarrega de descobrir como obter sua solução.



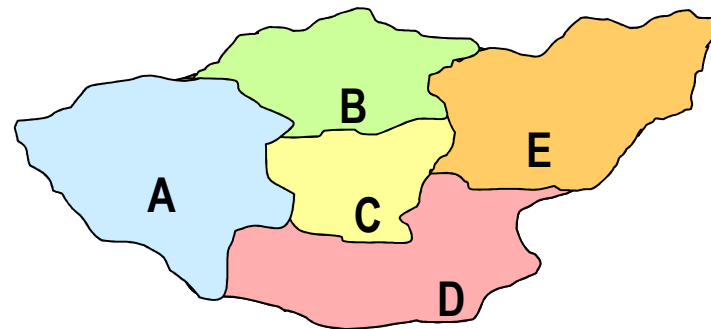
Exemplo – Coloração de mapas

- **Problema:** como colorir um mapa, usando no máximo quatro cores, de modo que regiões adjacentes tenham cores distintas?

- **Solução:**

- Primeiro, declaramos as cores que podem ser usadas na coloração; isto é feito por meio de sentenças denominadas **fatos**.

- Em seguida, declaramos que a tupla (A, B, C, D, E) , cujos componentes correspondem às regiões do mapa, é uma coloração válida se cada um de seus componentes é uma cor e se componentes representando regiões adjacentes no mapa têm valores distintos; isto é feito por meio de uma sentença denominada **regra**.





Exemplo – Coloração de mapas

- **Implementação:** definir os fatos e a regra na linguagem Prolog

```
% colorir.pl – coloração de mapas
```

```
% fatos
```

```
cor(azul).
```

```
cor(verde).
```

```
cor(amarelo).
```

```
cor(vermelho).
```

```
% regra
```

```
coloração(A,B,C,D,E) :-
```

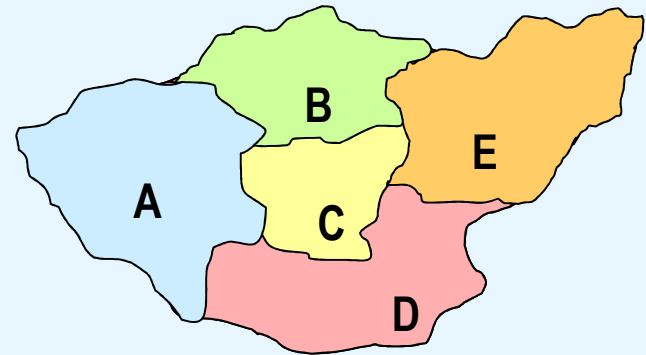
```
    cor(A), cor(B), cor(C), cor(D), cor(E),
```

```
    A\=B, A\=C, A\=D,
```

```
    B\=C, B\=E,
```

```
    C\=D, C\=E,
```

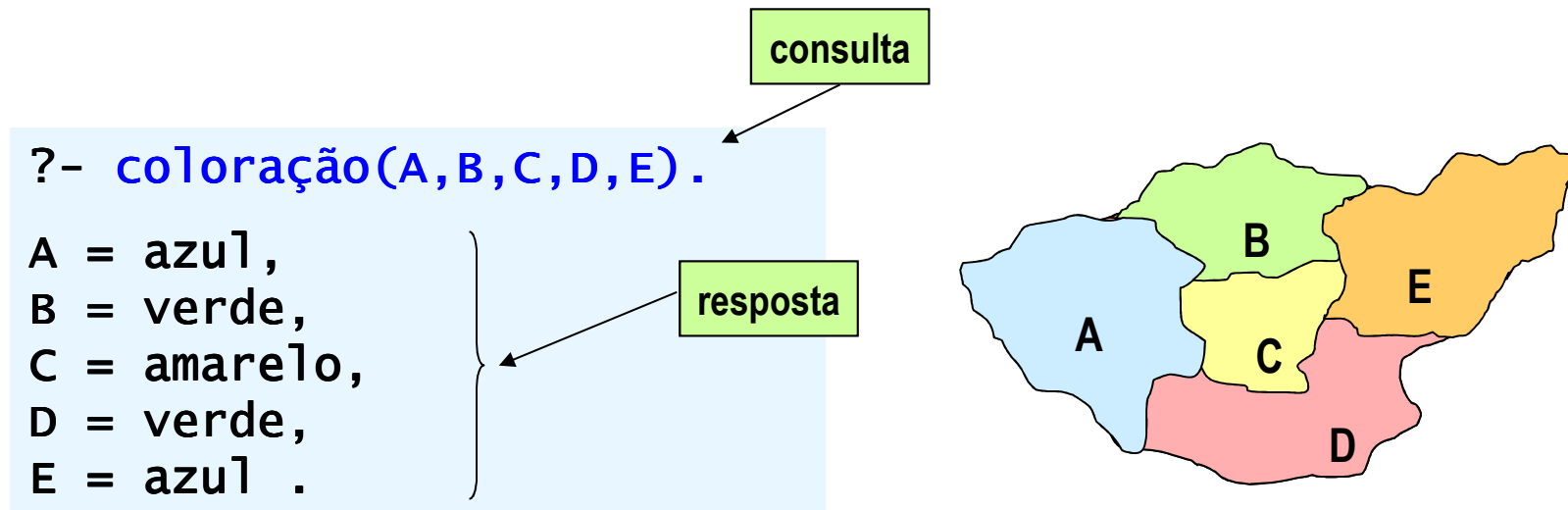
```
    D\=E.
```





Exemplo – Coloração de mapas

- **Teste:** após a compilação do programa, podemos consultar o Prolog para que ele encontre uma solução para o problema especificado.



- **Com este exemplo, temos a impressão de que o Prolog é inteligente.**
- **Como será que ele foi capaz de encontrar a solução do problema?**

Fim

