# GSI010 - Programação Lógica
# Resolvendo problemas com Programação Lógica

# Nesta aula

- Busca no espaço de estados

# Busca no espaço de estados

## Método

Considerar a existência de um **agente**, que possui **ações** para modificar os **estados** do mundo.

Simplificações

- o **estado** do mundo só muda quanda o **agente** faz uma **ação**
- meta do agente é fazer o mundo atingir um **estado** pré-definido

# Exemplo: O mundo do aspirador

## Agente: aspirador-robô

Qual é a melhor forma de aspirar sujeira?

## Ações

- entrarSala1
- entrarSala2
- aspirar

## Estados do mundo

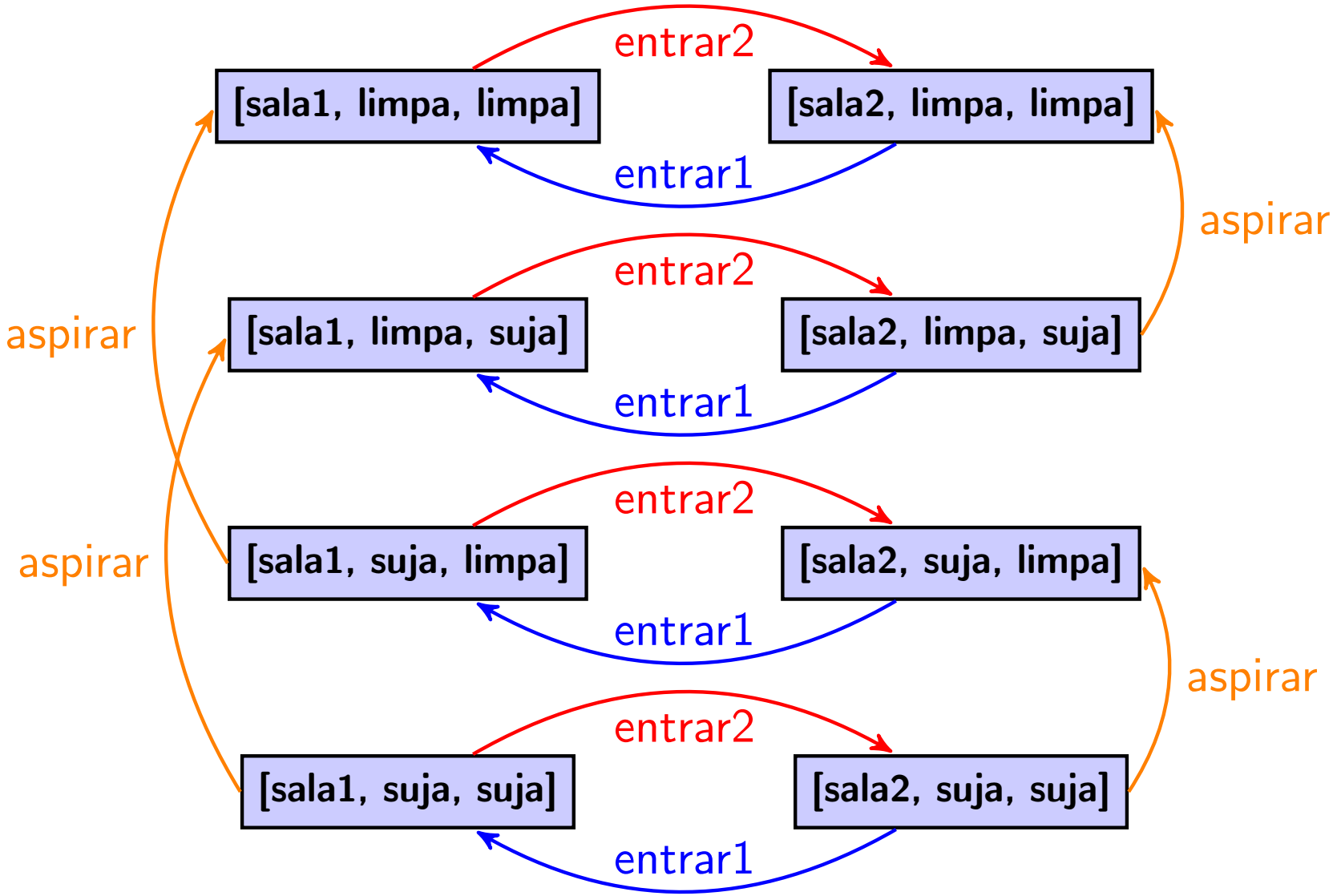
- sala 1 suja
- sala 1 limpa
- sala 2 suja
- sala 2 limpa

# Espaço de estados

Um grafo de estados tem

- conjunto de estados (nós/vértices)
- conjunto de ações - transições entre estados - arestas

# Exemplo

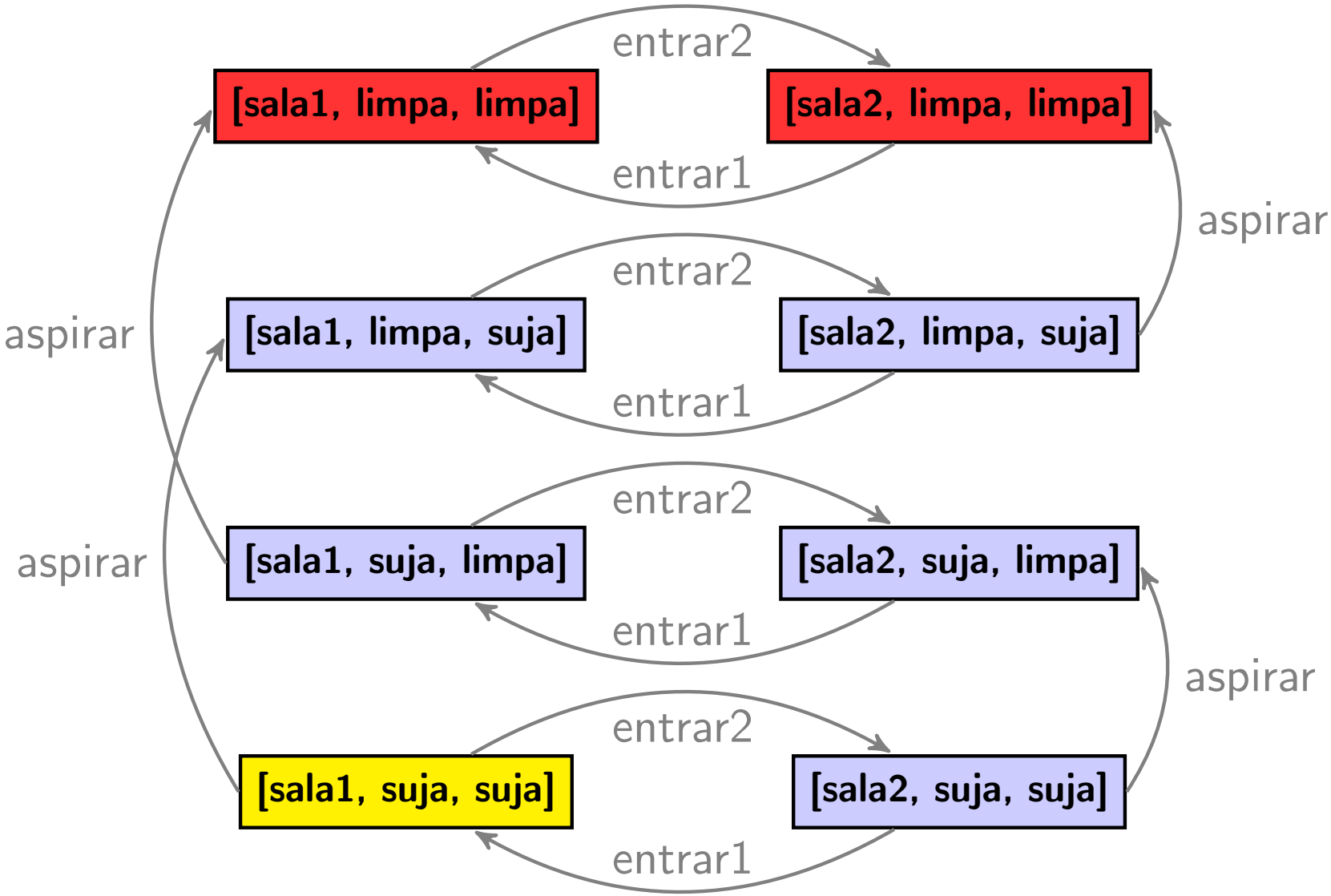

[sala1, limpa, limpa]
[sala2, limpa, limpa]
[sala1, limpa, suja]
[sala2, limpa, suja]
[sala1, suja, limpa]
[sala2, suja, limpa]
[sala1, suja, suja]
[sala2, suja, suja]
entrar2
entrar1
aspirar

# Espaço de estados

## Problema deve tem definido

- estado inicial
- estados finais - meta do agente

# Exemplo: estados inicial e finais

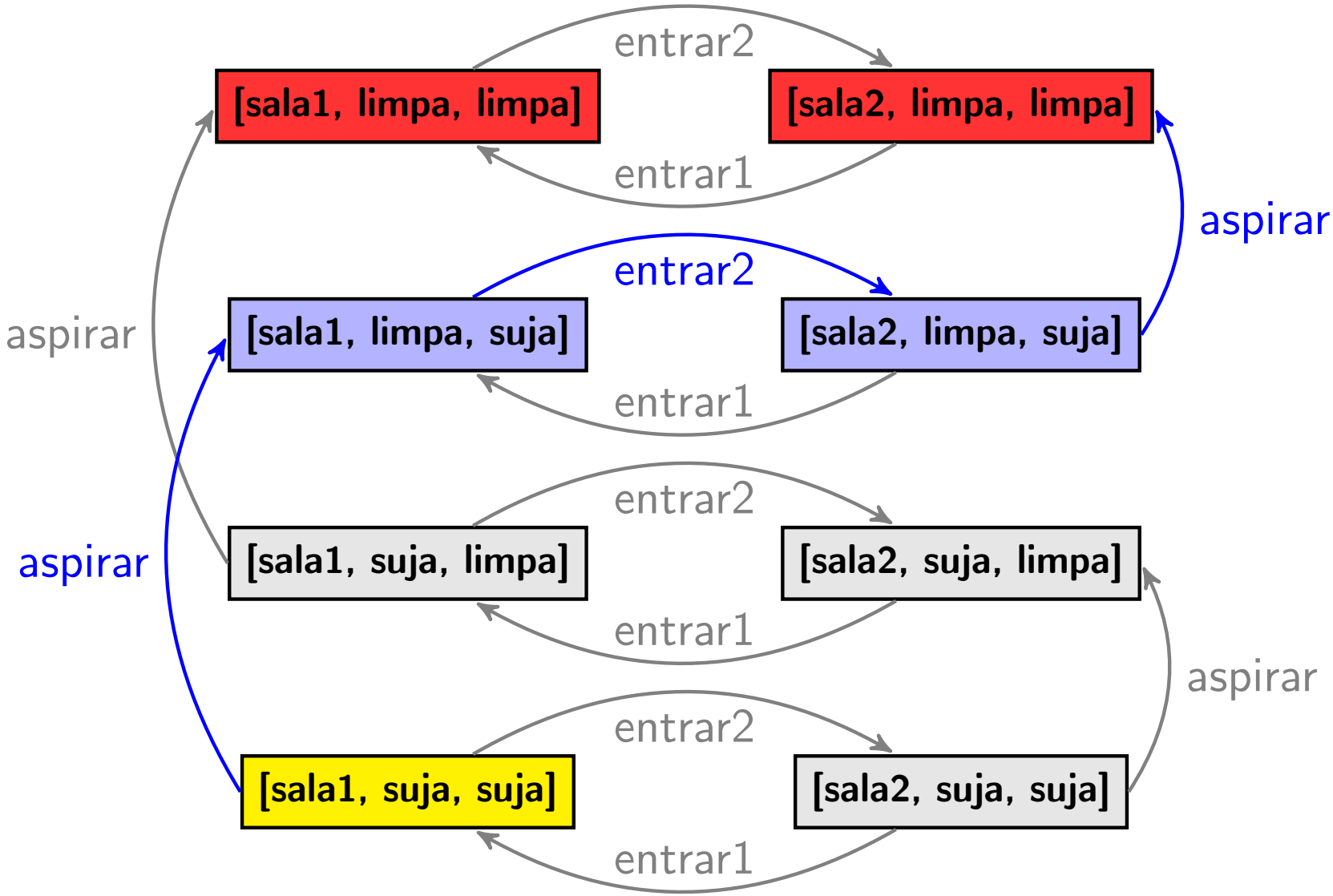


# Espaço de estados

## Solução do problema

- sequência de ações que quando executadas fazem o mundo sair do estado inicial e chegar em um estado final

# Solução do problema



# Representação de estados

## Estados são representados por estruturas

Estrutura de estados do Aspirador-robô

- lista com três elementos
  - [SalaAgente, Sala1Estado, Sala2Estado]
  - SalaAgente pode ser `sala1` ou `sala2`
  - Sala1Estado pode ser `limpa` ou `suja`
  - Sala2Estado pode ser `limpa` ou `suja`

## Exemplo: representação

```
estado_inicial([sala1, suja, suja]).
meta([_,limpa,limpa]).
```

# Ações – como representar ações?

## Predicado

```
acao(Acao, EstadoAnterior, EstadoPosterior) :- condicoes, efeitos.
```
1

- `Acao` – ação sendo realizada
- `EstadoAnterior` – estado do mundo atual
- `EstadoPosterior` – estado do mundo após a `Acao`
- `condicoes, efeitos` – pré-condições de realização da `Acao` e `efeitos` resultantes dela

# Exemplo: ação aspirar

```
acao(Acao, EstadoAnterior, EstadoPosterior) :- condicoes, efeitos.
```

## Ação aspirar

```
acao(aspirar, [SalaAntes, Sala1antes, Sala2antes],
              [SalaDepois, Sala1depois, Sala2depois]):-
 SalaAntes = sala1, Sala1antes = suja, %condicoes
 SalaDepois = sala1, Sala1depois = limpa, Sala2antes = Sala2depois. %efeitos

acao(aspirar, [SalaAntes, Sala1depois, Sala2antes],
              [SalaDepois, Sala1depois, Sala2depois]):-
 SalaAntes = sala2, Sala2antes = suja,
 SalaDepois = sala1, Sala1antes = Sala1depois, Sala2depois = limpa.
```

# Representação de ações

## Ação aspirar

```
acao(aspirar, [SalaAntes, Sala1antes, Sala2antes],                              1
              [SalaDepois, Sala1depois, Sala2depois]):-                         2
 SalaAntes = sala1, Sala1antes = suja, %condicoes                               3
 SalaDepois = sala1, Sala1depois = limpa, Sala2antes = Sala2depois. %efeitos    4
                                                                                5
```

## Ações: resumido

```
acao(entrar1,[sala2, S1, S2], [sala1, S1, S2]).                                 1
acao(entrar2,[sala1, S1, S2], [sala2, S1, S2]).                                 2
acao(aspirar,[sala1, suja, S2], [sala1, limpa, S2]).                            3
acao(aspirar,[sala2, S1, suja], [sala1, S1, limpa]).                            4
                                                                                5
```

# Ações aplicáveis

## Aplicável

Uma ação A é aplicável a um estado E se suas precondição são satisfeitas para esse estado.

## Encontrar ações aplicáveis: exemplo

```
?- findall(A, acao(A, [sala2, limpa, suja], _), Acoes).
  Acoes = [entrar1, aspirar].
```

# Estados sucessores

## Estados sucessores

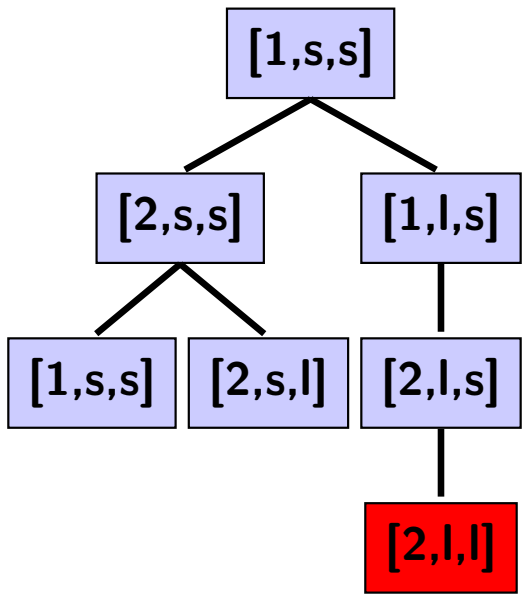
Estado alcançável por meio de apenas uma ação.

```
?- findall(E, acao(_, [sala2, limpa, suja], E), Estados).
Estados = [[sala1, limpa, suja], [sala2, limpa, limpa]].
```

# Árvore de busca

## Árvore de busca

Desdobramento das possibilidades de estados obtidos como efeito da aplicação de sequências de ações



# Estratégias de busca

## Estratégia descreve

Como fazer para encontrar solução na árvore de busca.

## Opções de estratégias

- busca aleatória
  - escolhe aleatoriamente para onde ir
- busca em largura
  - explora as possibilidades "um nível por vez"
- busca em profundidade
  - explora as possibilidades uma sequência por vez
- busca pelo menor custo
  - usa informação de avaliação das ações realizadas
- busca pela melhor estimativa
  - usa estratégia para avaliar aproximadamente boas alternativas
- busca ótima $A^*$
  - busca melhor solução possível

# Referências


- Slides – Busca no espaço de estados (Parte I) – Prof. Dr. Silvio do Lago Pereira