

Revisão G1

Reforçar os conteúdos abordados até o momento.

Márcio Nicolau

2025-09-25

Table of contents

Introdução	1
Estrutura da Revisão	2
Seção I: Questões de Múltipla Escolha	2
Questão 1	2
Questão 2	2
Questão 3	3
Seção II: Questões Dissertativas	3
Questão 4	3
Questão 5	4
Questão 6	5
Seção III: Questões Práticas em Python	5
Questão 7	5
Questão 8	6
Orientações Gerais	8
Referências Bibliográficas	8

List of Figures

Introdução

Este conjunto de atividades de revisão foi elaborado para consolidar o aprendizado para G1 de Inteligência Artificial.

Estrutura da Revisão

Esta revisão está organizada em três seções:

1. **Questões de Múltipla Escolha (3 questões)** - Cada questão com 5 alternativas (a-e)
2. **Questões Dissertativas (3 questões)** - Questões abertas que exigem explicação detalhada
3. **Questões Práticas em Python (2 questões)** - Implementação de algoritmos e conceitos

Seção I: Questões de Múltipla Escolha

Questão 1

Considerando as quatro abordagens principais para a definição de Inteligência Artificial propostas por Russell & Norvig, qual alternativa apresenta a combinação CORRETA de abordagem e sua característica principal?

Dica

Revise as quatro abordagens estudadas na Aula 2. Lembre-se de que elas se organizam em duas dimensões: “**pensar vs. agir**” e “**como humanos vs. racionalmente**”. Cada abordagem tem características específicas:

- Uma foca no comportamento observável (Teste de Turing)
- Uma foca na modelagem cognitiva interna
- Uma foca na lógica formal e “leis do pensamento”
- Uma foca na racionalidade e maximização de desempenho

- a) **Agir como Humanos** - Foca no uso de lógica formal e silogismos para tomar decisões corretas
- b) **Pensar como Humanos** - Busca replicar os mecanismos internos do pensamento humano através de modelagem cognitiva
- c) **Pensar Racionalmente** - Prioriza a maximização da utilidade esperada em ambientes estocásticos
- d) **Agir Racionalmente** - Baseia-se exclusivamente no Teste de Turing para validação de inteligência
- e) **Pensar como Humanos** - Utiliza apenas redes neurais para simular o comportamento cerebral

Questão 2

Em relação aos tipos de agentes inteligentes e suas características, analise as afirmações abaixo:

Dica

Analise cada afirmação individualmente considerando as características específicas de cada tipo de agente estudado na Aula 5. Lembre-se:

- Agentes reativos simples têm limitações sobre memória
- Agentes baseados em metas fazem planejamento, mas isso não garante otimalidade
- Agentes de aprendizado têm uma arquitetura específica com 4 componentes
- Agentes baseados em utilidade lidam com trade-offs entre objetivos

I. Um Agente Reativo Simples toma decisões baseadas apenas na percepção atual, sem considerar o histórico

- II. Um Agente Baseado em Metas sempre encontra a solução ótima para qualquer problema
- III. Um Agente de Aprendizado possui quatro componentes: Elemento de Aprendizado, Crítico, Gerador de Problemas e Elemento de Desempenho
- IV. Um Agente Baseado em Utilidade considera trade-offs entre objetivos conflitantes

Estão CORRETAS as afirmações:

- a) I e II apenas
- b) I, III e IV apenas
- c) II e IV apenas
- d) I, II e III apenas
- e) Todas as afirmações

Questão 3

Sobre algoritmos de busca (BFS e DFS), considere as seguintes proposições:

i Dica

Revise as características fundamentais dos algoritmos estudados na Aula 10:

- **BFS:** Qual estrutura de dados utiliza? (Fila ou Pilha?)
- **Otimidade:** BFS garante o caminho mais curto quando?
- **Memória:** Qual algoritmo é mais eficiente em termos de espaço?
- **Completeness:** Qual algoritmo pode ficar preso em loops infinitos?

I. A Busca em Largura (BFS) garante encontrar o caminho mais curto em número de passos quando os custos das ações são uniformes

II. A Busca em Profundidade (DFS) é mais eficiente em termos de memória que a BFS

III. A BFS utiliza uma pilha (stack) como estrutura de dados principal

IV. A DFS pode ficar presa em caminhos infinitos se não houver controle de visitação

Estão CORRETAS:

- a) I, II e III apenas
- b) I, II e IV apenas
- c) II, III e IV apenas
- d) I, III e IV apenas
- e) Todas as proposições

Seção II: Questões Dissertativas

Questão 4

Raciocínio Dedutivo vs. Indutivo em Inteligência Artificial

i Dica

Para o item (a): Pense na direção do raciocínio - um vai do geral para o específico, outro do específico para o geral. Considere também a certeza: um oferece conclusões necessárias (se premissas verdadeiras), outro oferece conclusões prováveis.

Para o item (b): Sistemas especialistas usam regras SE-ENTÃO (que tipo de raciocínio?). Machine Learning aprende padrões de dados (que tipo?).

Para o item (c): Como funciona o treinamento em ML? Você parte de exemplos específicos para criar modelos que generalizam?

- Defina** raciocínio dedutivo e raciocínio indutivo, explicando a principal diferença entre eles em termos de direção do raciocínio e certeza das conclusões.
- Forneça um exemplo prático** de cada tipo de raciocínio aplicado em sistemas de IA, explicando como cada abordagem seria utilizada.
- Análise** qual tipo de raciocínio é predominante no Aprendizado de Máquina moderno e **justifique** sua resposta com base nas características desse campo.

Questão 5

Classificação de Ambientes e Design de Agentes

Você foi contratado para projetar um agente inteligente para controlar um drone de entrega em uma cidade.

i Dica

Para o item (a): Pense nas 6 dimensões estudadas na Aula 6:

- **Observabilidade:** O drone consegue ver tudo? E atrás de prédios?
- **Determinismo:** O vento, outros drones, pedestres são previsíveis?
- **Episódico/Sequencial:** Uma entrega afeta a próxima?
- **Estático/Dinâmico:** O ambiente da cidade muda enquanto o drone “pensa”?
- **Discreto/Contínuo:** Posição e velocidade do drone são valores...
- **Agentes:** Há outros drones, carros, pedestres?

Para o item (b): Modelo PEAS - o que o drone deve maximizar (Performance)? Onde opera (Environment)? Como age (Actuators)? Como percebe (Sensors)?

Para o item (c): Considere se o drone precisa de planejamento, se deve aprender, se precisa lidar com incertezas...

- Classifique** o ambiente deste drone segundo as 6 dimensões estudadas (Observabilidade, Determinismo, Episódico/Sequencial, Estático/Dinâmico, Discreto/Contínuo, Único Agente/Multiagente). **Justifique** cada classificação.
- Descreva** o ambiente usando o modelo PEAS (Performance, Environment, Actuators, Sensors).
- Determine** qual tipo de agente inteligente (entre os 5 tipos estudados) seria mais apropriado para esta tarefa e **explique** por que, considerando as características do ambiente identificadas.

Questão 6

Lógica Proposicional vs. Lógica de Predicados

i Dica

Para o item (a): Em LP, como você representaria “todos os estudantes”? Precisaria de uma proposição para cada estudante individual? E se fossem 1000 estudantes?

Para o item (b): Em LPO, você pode usar quantificadores. Que quantificador expressa “todos”? Como representar propriedades como “ser estudante”, “estudar”, “passar na prova”?

Para o item (c): Lembre-se das regras: Instanciação Universal ($\forall x P(x) \rightarrow P(c)$) e Modus Ponens. Quais são os passos sequenciais?

- Explique** por que a Lógica Proposicional é inadequada para representar a sentença “Todos os estudantes que estudam passam na prova” de forma eficiente em um domínio com muitos estudantes.
- Traduza** a sentença acima para Lógica de Predicados de Primeira Ordem, definindo claramente os predicados utilizados.
- Demonstre** como seria possível inferir “João passou na prova” se tivéssemos na base de conhecimento a regra acima e o fato “João é estudante e João estudou”, utilizando as regras de inferência da LPO (Instanciação Universal e Modus Ponens Generalizado).

Seção III: Questões Práticas em Python

Questão 7

Sistema de Diagnóstico com Raciocínio Dedutivo

Implemente um sistema de diagnóstico médico simples usando raciocínio dedutivo. O sistema deve:

1. Ter uma base de regras predefinidas (mínimo 5 regras)
2. Receber uma lista de sintomas como entrada
3. Aplicar as regras dedutivamente para chegar a diagnósticos
4. Retornar todos os diagnósticos possíveis e recomendações

i Dica

Estrutura das regras: Use dicionários ou tuplas para representar regras SE-ENTÃO:

- `{'condicoes': ['sintoma1', 'sintoma2'], 'conclusao': 'diagnostico'}`

Aplicação das regras: Para cada regra, verifique se todas as condições estão presentes nos sintomas de entrada.

Raciocínio dedutivo: Lembre-se do exemplo da Aula 3 - se as premissas são verdadeiras e a regra é válida, a conclusão DEVE ser verdadeira.

Regras compostas: Uma regra pode usar como condição o resultado de outras regras (ex: “SE gripe E febre_alta ENTÃO repouso”).

Requisitos técnicos:

- Use estruturas de dados adequadas para representar regras
- Implemente a aplicação de regras de forma clara
- Inclua pelo menos uma regra que combine dois diagnósticos para uma nova conclusão
- Teste o sistema com pelo menos 3 casos diferentes

```
def sistema_diagnostico_dedutivo(sintomas):
    """
    Sistema de diagnóstico que usa raciocínio dedutivo para inferir
    diagnósticos a partir de sintomas.

    Args:
        sintomas (list): Lista de sintomas observados

    Returns:
        list: Lista de diagnósticos e recomendações
    """
    # Sua implementação aqui
    pass

# Exemplo de uso esperado:
# sintomas_paciente = ['febre', 'tosse', 'dor_de_garganta']
# resultado = sistema_diagnostico_dedutivo(sintomas_paciente)
# print(f"Diagnósticos: {resultado}")
```

Estruture sua resposta incluindo:

- Definição das regras (explicite a lógica de cada uma)
- Implementação completa da função
- Exemplos de teste com saída esperada
- Breve explicação de como o raciocínio dedutivo está sendo aplicado

Questão 8

Comparação Prática de BFS e DFS

Implemente e compare os algoritmos de Busca em Largura (BFS) e Busca em Profundidade (DFS) para encontrar caminhos em um grafo representando uma rede de transporte urbano.

Especificações:

1. Implemente ambos os algoritmos com capacidade de rastreamento de visitação
2. O grafo deve representar pelo menos 8 estações conectadas
3. Compare os algoritmos em termos de:
 - Ordem de exploração dos nós
 - Caminho encontrado
 - Número de nós expandidos
 - Adequação para o problema

i Dica

Estruturas de dados:

- **BFS:** Use `collections.deque()` - adicione com `append()`, remova com `popleft()` (FIFO)
- **DFS:** Use lista Python - adicione com `append()`, remova com `pop()` (LIFO)

Rastreamento: Mantenha listas/contadores para:

- Nós visitados (conjunto `set()`)
- Ordem de exploração (lista)
- Caminho até cada nó

Análise: Compare qual encontra caminho mais curto, qual usa menos memória, qual é mais rápido para este grafo específico.

Lembre-se: BFS explora “nível por nível”, DFS vai “fundo primeiro”.

```
from collections import deque

def bfs_com_analise(grafo, inicio, objetivo):
    """
    Busca em Largura com análise detalhada do processo.

    Returns:
        tuple: (caminho_encontrado, nos_expandidos, ordem_expansao)
    """
    # Sua implementação aqui
    pass

def dfs_com_analise(grafo, inicio, objetivo):
    """
    Busca em Profundidade com análise detalhada do processo.

    Returns:
        tuple: (caminho_encontrado, nos_expandidos, ordem_expansao)
    """
    # Sua implementação aqui
    pass

def comparar_algoritmos(grafo, inicio, objetivo):
    """
    Compara BFS e DFS e apresenta análise detalhada.
    """
    # Sua implementação aqui
    pass

# Exemplo de grafo de transporte urbano para teste
grafo_transporte = {
    'Estacao_Central': ['Estacao_Norte', 'Estacao_Sul', 'Estacao_Leste'],
```

```
'Estacao_Norte': ['Estacao_Central', 'Estacao_Nordeste'],
'Estacao_Sul': ['Estacao_Central', 'Estacao_Sudeste', 'Estacao_Sudoeste'],
'Estacao_Leste': ['Estacao_Central', 'Estacao_Nordeste'],
'Estacao_Nordeste': ['Estacao_Norte', 'Estacao_Leste'],
'Estacao_Sudeste': ['Estacao_Sul'],
'Estacao_Sudoeste': ['Estacao_Sul', 'Terminal_Oeste'],
'Terminal_Oeste': ['Estacao_Sudoeste']
}
```

Estruture sua resposta incluindo:

- Implementação completa de ambos os algoritmos
- Função de comparação que execute e analise ambos
- Teste executado de “Estacao_Central” para “Terminal_Oeste”
- Análise detalhada dos resultados explicando:
 - Qual algoritmo foi mais eficiente e por quê
 - Diferenças na ordem de exploração
 - Qual seria mais apropriado para diferentes cenários do transporte urbano

Orientações Gerais

1. **Tempo recomendado:** 3 horas
2. **Material de consulta:** Apenas anotações de aula e material do curso
3. **Questões práticas:** Teste seu código antes de entregar
4. **Organização:** Numere claramente suas respostas
5. **Dissertativas:** Seja objetivo mas completo em suas explicações

Boa sorte em sua revisão!

As questões desta revisão são baseadas nos conceitos apresentados nas aulas do curso, que por sua vez fundamentam-se em:

- Livro texto: ([Russell; Norvig, 2004](#))
- Material didático das aulas 1-10 da disciplina

Referências Bibliográficas

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial: Um Enfoque Moderno**. 2. ed. Rio de Janeiro: Prentice Hall, 2004.