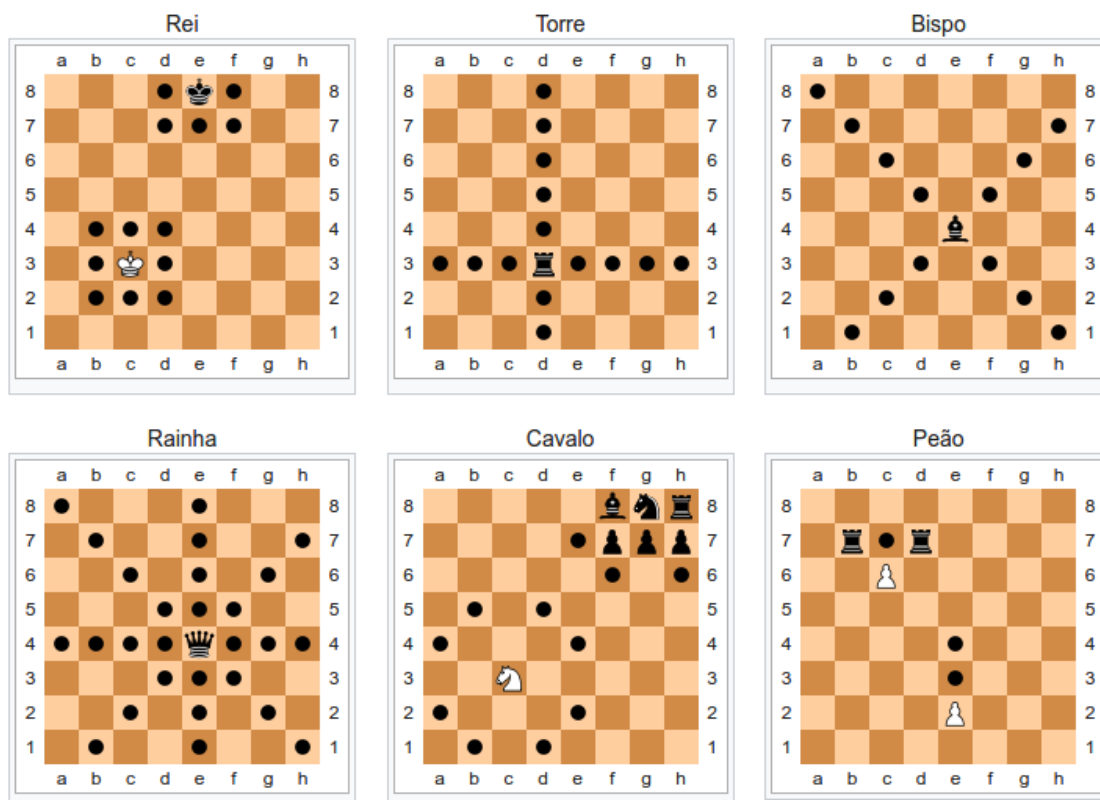


Linguagens de Programação

Problemas para Resolução em Linguagem Lógica (Prolog)

1. Construa um conjunto de regras para validar os movimentos de todas as peças de um jogo de xadrez. Em um jogo de xadrez existem 6 tipos de peças diferentes: Peão, Torre, Bispo, Cavalo, Rainha e Rei.
 - (a) O tabuleiro é composto por uma matriz 8x8, totalizando 64 posições. Crie fatos para posicionar todas as peças do jogo. Cada fato deve vincular o tipo da peça, um identificador, a cor, a posição horizontal e vertical da peça. Por exemplo: `posicao(torre, 1, branco, 8, 1)`. Nesse exemplo a torre 1 branca (o numeral 1 é o identificador da torre, já que existe mais de uma torre branca) será posicionada na casa h1.
 - (b) Os movimentos das peças são como apresentados nas imagens abaixo.



1. A **Torre** se movimenta nas direções ortogonais, isto é, pelas linhas (horizontais) e colunas (verticais), não podendo se mover pelas diagonais. Ela pode mover quantas casas desejar pelas colunas e linhas, porém, apenas em um sentido em cada jogada.
2. O **Bispo** se movimenta nas direções diagonais, ou seja, na direção das casas da mesma cor. Ele pode mover quantas casas desejar pelas diagonais, porém, apenas em um sentido (cada jogada), existe o bispo da casa preta e o bispo da casa branca, e os mesmos não podem mudar de cor durante o jogo.
3. A **Rainha** é a peça mais poderosa, pois pode movimentar-se quantas casas quiser ou puder, na diagonal, vertical ou horizontal, porém, apenas em um sentido em cada jogada, a dama (ou rainha) anda com os movimentos de todas as outras peças (exceto o cavalo), andando quantas casas quiser.
4. O **Rei** pode se mover em todas as direções somente uma casa de cada vez, desde que o movimento não seja para uma casa ameaçada por uma peça adversária. O rei também pode capturar qualquer peça adversária, desde que a mesma não tenha outra peça defendendo-a. Um Rei nunca poderá dar xeque a outro Rei.
5. O movimento do **Cavalo** é em "forma de L", ou seja, anda duas casas na horizontal ou vertical e depois uma casa na vertical ou horizontal, ou vice-versa. O cavalo pode saltar sobre qualquer peça sua ou do adversário. A captura ocorre quando uma peça adversária se encontra na casa final do movimento realizado pelo cavalo.
6. O **Peão** move-se em coluna (vertical) somente para a frente e uma casa, nunca para trás. Quando um peão alcança a última fileira do tabuleiro (fileira 8 para as brancas ou 1 para as pretas) ele é promovido, tornando-se uma Torre, Bispo, Cavalo ou Dama, conforme o desejo do jogador.

(fonte: https://pt.wikipedia.org/wiki/Leis_do_xadrez)

- (c) Crie regras para indicar as possíveis próximas posições de cada peça. Certifique-se de que a regra está funcionando corretamente fazendo algumas consultas. Por exemplo: **proxima_jogada**(torre, 1, branca, X, Y). Nesse exemplo X receberia a posição horizontal e Y a vertical. Para uma consulta como essa, devem ser retornadas todas as possibilidades de movimentação.

1. Dica: será necessário adicionar uma regra para cada tipo de peça.

- (d) Incremente a regra `proxima_jogada` para levar em conta (e descartar) as posições que já estão ocupadas com outras peças.
- (e) Adicione a regra **`proxima_jogada_ataque`** que, quando utilizada em consulta, pode retornar as posições para ataque válidas. A regra deve levar em conta o movimento de ataque e as peças da cor adversária que estão em tais posições. A consulta poderá ser feita como no exemplo: `proxima_jogada_ataque(torre, 1, branca, X, Y)`. Nesse caso, só serão retornadas as possibilidades de ataque (posições que essa peça pode se movimentar e que possuem peças inimigas).

2. Suponha que em uma rede social existem os conceitos: usuário, interesse e grupo. Crie fatos e regras para:

- (a) Definir 20 **usuários**, 10 **interesses** e 5 **grupos**.
- (b) Definir os conceitos **`seguir`** e **`amigo`**. Um usuário pode seguir outro usuário. Dados os usuários A e B, quando A segue B e B segue A então diz-se que A e B são amigos.
- (c) Definir relacionamentos **`seguir`** entre todos os usuários, de modo que ao menos 5 sejam amigos entre si. Valide a regra **`amigo`** através de consultas.
- (d) Um usuário pode **`gostar`** de algum interesse. Defina fatos que relacionem ao menos 2 interesses com cada usuário.
- (e) Um usuário pode pertencer a um grupo. Se um usuário **`pertence`** a um grupo, o grupo **`contém`** o usuário. Defina fatos para adicionar ao menos 5 usuários em cada grupo. Realize consultas com `pertence` e `contém` para validar as regras.
- (f) Crie uma consulta que retorne todos os interesses dos amigos de um determinado usuário.
- (g) Crie uma consulta que retorne todos os amigos de um determinado usuário que pertencem a um determinado grupo.
- (h) Crie uma consulta que, dado um grupo, retorne todos os interesses de todos os usuários daquele grupo.
- (i) Crie uma consulta que indique para uma pessoa novos interesses a partir das seguintes regras:
 - 1. os interesses dos amigos daquela pessoa que ela ainda não possui OU
 - 2. os interesses das pessoas que pertencem aos mesmos grupos daquela pessoa e que ela ainda não possui.