Programação 3
Trabalho Conversor de Notação Algébrica para xadrez
Por: Carlos Palma 46520

Introdução

Foi proposto pelo professor Salvador Abreu responsável pela cadeira de programação 3 a elaboração de um conversor de notação para jogos de xadrez. Poderíamos escolher a notação que preferíamos entre Algébrica, Descritiva e Postal bem como a linguagem de programação a utilizar entre Prolog, CLP ou Ocaml. Na minha ótica escolhi a notação algébrica por ser a que melhor entendo derivado do facto de ser a mais utilizada em jogos de xadrez e como linguagem escolhi Ocaml por ser a linguagem mais fresca na minha mente devido ao facto de ter sido a que estudei para o mini teste 3. O nosso programa deveria converter um input recebido através de um ficheiro de texto para jogadas de Xadrez, representando um tabuleiro e imprimindo o mesmo da tela sempre que ocorria uma jogada.

Desenvolvimento

Comecei por tentar perceber aquilo que era a notação álgebrica e de que forma funciona a mesma, rapidamente percebi que o input apenas nos dava a posição final das peças da jogada de cada jogador e ai constava a dificuldade do problema pois teria que calcular a partir da posição final e tipo de peça qual a peça que conseguiria mover e se existia sequer uma peça que poderia mover.
Comecei por tratar do input, para mim o mais conveniente seria tratar o input como uma lista derivado do facto de termos passado a grande maioria do semestre a trabalhar com listas, dessa forma comecei por desenvolver a função **ler_jogadas** que recebe uma string file com o nome de um ficheiro e retorna uma lista de strings com as jogadas contidas no mesmo. Ela faz isso abrindo o ficheiro em um canal de leitura, e então usa uma função recursiva chamada loop para ler cada linha do ficheiro e adicioná-la à lista de jogadas. O corpo da função loop tenta ler uma linha do ficheiro usando a função input_line e, em seguida, verifica se a linha lida é uma string vazia. Se for, a função loop retorna a lista de jogadas invertida usando a função List.rev. Se a linha não for vazia, a função divide a linha em uma lista de palavras usando um loop para percorrer cada caractere da linha e verificar se é um espaço ou uma tabulação. Em seguida, a função adiciona essas palavras à lista de jogadas e chama a si mesma novamente para ler a próxima linha do ficheiro. Se o fim do ficheiro for encontrado durante a leitura, a função loop retorna a lista de jogadas invertida. De seguida para tratar cada jogada percebi que a melhor forma a meu ver seria verificar a primeira letra de cada string na lista e consoante a mesma direcionar o input para outra função que trataria do mesmo da forma suposta, dessa forma desenvolvi função **check_input** que recebe uma lista de strings lst e retorna uma nova lista de strings após aplicar uma função específica a cada elemento da lista. A função é determinada pelo primeiro caractere de cada string da lista. A função usa a função List.mapi para aplicar uma função anônima a cada elemento da lista, junto com seu índice. O primeiro caractere da string é obtido usando String.get s 0, e então verifica se é um dígito, uma letra minúscula ou uma letra maiúscula usando as funções Char.code e Char.lowercase_ascii. Dependendo do tipo de caractere, a função check_input chama uma das funções process_digit, process_lowercase ou process_uppercase passando a string e o índice atuais como argumentos.
Seria então agora desenvolvido o passo mais importante, codificar funções que fossem responsáveis por verificar a legalidade de cada jogada e apresentar ao utilizador cada jogada sempre que assim o fosse possível, comecei por desenvolver uma matriz 8 por 8 de forma a representar o tabuleiro inicializando a mesma com peças nas posições correspondentes ao inicio de um jogo de xadrez, utilizei letras maiusculas e minusculas para diferencia as peças de cada jogador(minusculas→peões, maisculas→Outra peça consoante a primeira letra, digitos →condição a acontecer) e desenvolvi uma função print_board muito simples que simplesmente percorre o tabuleiro e vai printando as peças. Iniciei então o desenvolvimento da função relativa às jogadas legais do peão e rapidamente

percebi que existia a necessidade de transformar os inputs em algo que seria possível utilizar numa matriz, como tal escrevi as funções mapping e letter_to_number, sendo que mapping é utilizada para mapear as letras do input na coluna correspondente da matriz e letter_to_number é utilizada para realizar a conversão das mesmas. Após o input completamente tratado comecei finalmente o desenvolvimento da função responsável por verificar a legalidade de um movimento de um peão, o que a meu ver até foi bastante simples, sendo que os peões só se podem mover para a frente 1 casa ou 2 casas se for a jogada inicial, simplesmente verifiquei se a partir da posição final dada como input existia um peão nas duas casas anteriores e em caso positivo verificava também se essas posições não estavam ocupadas por qualquer peça. Verifiquei também em caso do segundo caracter da string ser um x se seria possível a algum peão mover-se na diagonal para a posição final e desta forma "comer" alguma das peças.

Estava tudo a correr bem foi então que passei para o desenvolvimento dos movimentos legais das outras peças e ai começaram a surgir problemas… comecei a desenvolver a torre e rapidamente percebi que a forma como estava a desenvolver o trabalho seria insustentável, para verificar se seria possível mover uma torre para a posição final teria de verificar se existia uma torre primeiramente em todo o tabuleiro na vertical/horizontal e após descobrir a posição dessa torre seria necessário verificar se em todo o espaço entre a torre e a posição final não existia qualquer tipo de peças no caminho, tentei codificar estes problemas no entanto após varias tentativas percebi que seria super difícil, apenas com cadeias de if's e for's e então percebi que não estava a conseguir desenvolver o problema da forma correta, foi então que decidi enviar e-mail ao professor que me indicou aquilo que suspeitava, não estava a abordar o problema da forma correta! Resolvi então iniciar uma abordagem de desenvolvimento com recurso a type's no entanto apenas com 4 dias para desenvolver todo um novo trabalho e sendo type's a matéria com que menos me identificava, comecei por estudar toda a matéria e só depois iniciei a minha tentativa de elaboração do mesmo. No entanto as minhas inúmeras tentativas e abordagens não foram bem sucedidas. Tenho noção agora que a abordagem mais correta teria sido criar um tipo para a peça, um tipo para o movimento e utilizar um predicado de base de dados para guardar as posições no tabuleiro. Utilizaria funções para verificar quais os movimentos que poderiam ser realizados e em caso positivo utilizaria outras funções para atualizar o tabuleiro.

Conclusão

Tenho perfeita noção do quão pobre está o trabalho, entreguei o mesmo por recomendação do professor e na esperança que possa aprender algo para o futuro. Fiquei bastante triste e frustrado pela minha falta de capacidade no desenvolver do mesmo. Tentei acompanhar a matéria durante todo o semestre e esforcei-me para conseguir perceber aquilo que é uma linguagem de programação lógica no entanto agora percebo que não foi suficiente.

Compreendo se o professor não me quiser chamar para apresentação e apenas me enviar um e-mail com aspetos que me possam vir a ajudar no futuro.