

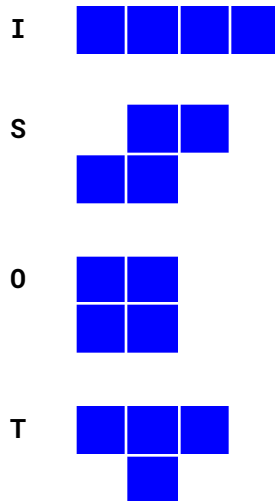
# Trabalho de Programação III

## 2023/24

### Paktris

#### Enunciado

Considere 4 peças do Tetris, com as seguintes designações e formas, suscetíveis de serem reorientadas (só por rotação, nunca por simetria):



1. Considere um tabuleiro de  $N \times M$  células (largura  $M$ , altura  $N$ ), o seu programa deverá admitir uma lista de jogadas, na forma de triplos (**PEÇA**, **NROT**, **NDIR**) em que:
  - **PEÇA** é o tipo de peça indicado (em OCaml poderá ser um tipo soma com valores **I**, **S**, **O** e **T**; em Prolog poderão ser os átomos **i**, **s**, **o** e **t**)
  - **NROT** é um inteiro que representa o número de vezes que a peça é rodada  $90^\circ$  no sentido dos ponteiros do relógio. Assume-se que a rotação é sempre realizada com o canto inferior esquerdo da peça mantido fixo (ou seja, "encosta" em baixo à esquerda).
  - **NDIR** é um inteiro que representa o número de posições que a peça é empurrada para a direita, antes de ser largada
  - $N$  e  $M$  são dados, respectivamente, por variáveis globais  $n$  e  $m$  no caso do OCaml, e por predicados  $n(N)$  e  $m(M)$  no caso do Prolog.

O programa deverá **validar se uma sequência** (lista) de jogadas (o seu parâmetro) **cabe no tabuleiro**, com as regras do Tetris.

Recordam-se as regras: depois de posicionada a peça, esta é largada e "cai" até onde bater noutra peça ou chegar ao fundo do tabuleiro. Após uma peça cair, o tabuleiro fica pronto para receber a próxima peça. Se uma peça ficar acima do topo do tabuleiro, considera-se que a sequência não cabe.

No caso do Prolog, o programa deverá ser implementado por um predicado **paktris(L)** em que **L** é uma lista de jogadas e que sucede ("yes") caso a sequência caiba e falha caso contrário. Exemplo:

```
% assume-se que n(4) e m(4)
| ?- paktris([(i,0,0), (i,0,0), (o,0,2), (o,0,0)]).
yes

| ?- paktris([(i,1,0), (s,1,1), (o,0,1)]).
no
```

No caso do OCaml, deverá ser uma função que admite uma lista de jogadas e retorna um booleano (**true** se as jogadas couberem, **false** caso contrário). Por exemplo:

```
(* assume-se que n=4 e m=4 *)
# paktris [(I,0,0); (I,0,0); (O,0,2); (O,0,0)];;
- : bool = true

# paktris [(I,1,0); (S,1,1); (O,0,1)];;
- : bool = false
```

2. Melhore o programa para visualizar o tabuleiro no final das jogadas (pode fazer gráficos ASCII ou usar qualquer outro método de visualização, por exemplo HTML).

## Componentes do trabalho

O trabalho consiste em:

1. Um programa (em Prolog ou OCaml) que implemente o pedido acima. O ficheiro com o programa deverá chamar-se **paktris.pl** ou **paktris.ml**.
2. Um relatório breve (2-4p) que:
  - a. Explique as opções tomadas (representação das jogadas, representação do tabuleiro, etc.)
  - b. Explique a organização do código

- c. Indique quais os casos que ficaram por tratar

## Organização

O trabalho deve ser realizado por **grupos de 1 ou 2 elementos**. Não são admitidos grupos com 3 ou mais elementos.

## Datas

Data limite de entrega: **2024.01.14**

Data para eventuais discussões: a marcar caso-a-caso, após a entrega