

Alunos:

Daniel dos Santos Barros de Sousa - 211030980

Marcos Gabriel Tavares - 170041042

Rodrigo Tiago Costa Lima - 180037242

Vinicius Vieira de Souza - 170115500

# Towers

## O jogo

Trata-se de um quebra-cabeça que consiste em uma base contendo três pinos, sendo que em um deles é disposto alguns discos, uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo. O objetivo é mover todos os discos para a torre de destino.

1. Regras:

- a. Só pode mover um disco por vez
- b. Um disco maior nunca pode ficar em cima de um disco menor

A melhor solução, possui a quantidade de movimentos calculada pela fórmula  $2^X - 1$ , onde X é o número de discos.

## Artigos Científicos

Há dois artigos acerca desse domínio, um específico, [HTN IPC-2020 Domains: Towers](#), e um geral, [Bound to Plan: Exploiting Classical Heuristics via Automatic Translations of Tail-Recursive HTN Problems](#), que inclusive é referenciado no README do domínio.

## Domínio em PDDL

Ambos possuem uma única ação, sendo ela a move, que move um anel para cima de outro anel ou torre. O domínio em PDDL performa muito melhor que o domínio em HDDL para a mesma tarefa de mover os anéis de uma torre para outra, apesar de não ter o conhecimento do domínio codificado em si.

# Domínio em HDDL

O domínio conta com as informações usuais na montagem de domínio, sendo elas:

**Tipos:** O domínio apresenta apenas um tipo, sendo o de OBJ (Objeto), que molda dois fatores no domínio.

- ❖ **RING** - Representa anéis que podem ser movidos entre as torres do problema, tendo propriedades específicas, como tamanho e posição em relação a outras peças.
- ❖ **TOWER** - Representa as torres, sendo que elas têm capacidade de empilhar anéis em uma ordem específica e seguir as restrições específicas do problema.

**Predicados (Condições):** O domínio tem algumas propriedades já definidas, sendo elas:

- ❖ **(on ?r - RING ?o - OBJ)** - Indica se um anel específico está em cima de um objeto.
- ❖ **(goal\_on ?r - RING ?o - OBJ)** - Indica se um anel específico deve estar em uma posição específica.
- ❖ **(towerTop ?o - OBJ ?t - TOWER)** - Indica se um objeto específico está no topo da torre específica.
- ❖ **(smallerThan ?r - RING ?o - OBJ)** - Indica se um anel específico é menor que um objeto específico.

**Tarefas:** Determina as tarefas disponíveis no domínio.

- ❖ **(:task shiftTower :parameters (?t1 - TOWER ?t2 - TOWER ?t3 - TOWER))** - Tarefa de mover uma torre específica para outra torre específica usando uma torre intermediária.
- ❖ **(:task selectDirection :parameters (?r - RING ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER))** - Tarefa de selecionar a direção para qual se deseja mover um anel específico em cima de uma torre específica.
- ❖ **(:task rotateTower :parameters (?t1 - TOWER ?t2 - TOWER ?t3 - TOWER))** - Tarefa de girar uma torre específica em direção a outra torre específica usando uma torre intermediária.
- ❖ **(:task exchange :parameters (?t1 - TOWER ?t2 - TOWER ?t3 - TOWER))** - Tarefa de trocar a posição duas torres de lugares usando uma torre intermediária.
- ❖ **(:task move\_abstract :parameters (?t1 - TOWER ?t2 - TOWER))** - Abstração do movimento entre duas torres.

**Métodos:** Determina os métodos das tarefas definidas.

- ❖ **m-shifttower**
  - **parameters:** (?r - RING ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (shiftTower ?t1 ?t2 ?t3)
  - **preconditions:** (towerTop ?r ?t1)
  - **ordered-tasks:** (and (selectDirection ?r ?t1 ?t2 ?t3)))
- ❖ **selectedDirection**
  - **parameters:** (?r - RING ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (selectDirection ?r ?t1 ?t2 ?t3)
  - **preconditions:** (on ?r ?t1)
  - **ordered-tasks:** (and (rotateTower ?t1 ?t3 ?t2)))
- ❖ **m-selectDirection**
  - **parameters:** (?r - RING ?r1 - RING ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (selectDirection ?r ?t1 ?t2 ?t3)
  - **preconditions:** (on ?r ?r1)
  - **ordered-tasks:** (and (selectDirection ?r1 ?t1 ?t3 ?t2)))
- ❖ **m-rotateTower**
  - **parameters:** (?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (rotateTower ?t1 ?t2 ?t3)
  - **ordered-tasks:** (and (move\_abstract ?t1 ?t2) (exchange ?t1 ?t2 ?t3)))
- ❖ **m-rotateClear**
  - **parameters:** (?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (rotateTower ?t1 ?t2 ?t3)
  - **preconditions:** (and (towerTop ?t1 ?t1) (towerTop ?t3 ?t3))
  - **ordered-tasks:** (and )
- ❖ **exchangeLR**
  - **parameters:** (?r1 - RING ?o3 - OBJ ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (exchange ?t1 ?t2 ?t3)
  - **preconditions:** (and (towerTop ?r1 ?t1) (towerTop ?o3 ?t3) (smallerThan ?r1 ?o3))
  - **ordered-tasks:** (and (move\_abstract ?t1 ?t3) (rotateTower ?t2 ?t3 ?t1)))
- ❖ **exchangeRL**
  - **parameters:** (?r1 - RING ?o3 - OBJ ?t1 - TOWER ?t2 - TOWER ?t3 - TOWER)
  - **task:** (exchange ?t1 ?t2 ?t3)
  - **preconditions:** (and (towerTop ?o1 ?t1) (towerTop ?r3 ?t3) (smallerThan ?r3 ?o1))
  - **ordered-tasks:** (and (move\_abstract ?t3 ?t1) (rotateTower ?t2 ?t3 ?t1)))
- ❖ **newMethod21**
  - **parameters:** (?r - RING ?o1 - OBJ ?t1 - TOWER ?o2 - OBJ ?t2 - TOWER)
  - **task:** (move\_abstract ?t1 ?t2)
  - **preconditions:** (and (towerTop ?o1 ?t1) (towerTop ?r3 ?t3) (smallerThan ?r3 ?o1))

➤ **ordered-tasks: (move ?r ?o1 ?t1 ?o2 ?t2)**

**Actions:** Ações que modificam os predicados, sendo que o domínio apresenta apenas uma.

- ❖ **move-** Move uma anel em cima de uma torre ou anel para cima de uma outra torre ou anel.
- ❖ **parameters** (?r - RING ?o1 - OBJ ?t1 - TOWER ?o2 - OBJ ?t2 - TOWER)
- ❖ **preconditions**
  - (and
  - (towerTop ?r ?t1)
  - (towerTop ?o2 ?t2)
  - (on ?r ?o1)
  - (smallerThan ?r ?o2))
- ❖ **effect:**
  - (and
  - (not (on ?r ?o1))
  - (on ?r ?o2)
  - (not (towerTop ?r ?t1))
  - (towerTop ?o1 ?t1)
  - (not (towerTop ?o2 ?t2))
  - (towerTop ?r ?t2)))

## Execução dos Problemas

A Tabela 1 possui o tempo da execução dos problemas utilizando o algoritmo vencedor da *Track Agile* (PandaDealer-agile-1) da IPC 2023 e o tempo da execução dos problemas utilizando o *alias* (seq-opt-fdss-2) do Fast-downward utilizando a versão em PDDL para comparação. O tempo limite de execução foi de 1800s.

Problem/Planner	HDDL Total-Order	PDDL OPT seq-opt-fdss-2	Numero de ações
	PandaDealer-agile-1	Fast-downward	
1	0.08s	0.209s	1
2	0.102s	0.157s	3
3	0.134s	0.159s	7
4	0.158s	0.207s	15
5	0.173s	0.202s	31
6	0.239s	0.220s	63
7	0.537s	0.244s	127
8	1.302s	0.273s	255
9	3.584s	0.315s	511
10	11.438s	0.502s	1023
11	39.021s	0.903s	2047
12	178.36s	1.930s	4095
13	695.639s	4.830s	8191
14	TLE	13.546s	16383
15	TLE	40.606s	32767
16	TLE	121.673s	65535
17	TLE	MLE	
18	TLE	MLE	
19	TLE	MLE	
20	TLE	MLE	
Total		13   16	

Tabela 1 - Comparação de HDDL e PDDL

## Exemplo de plano

Para exemplificar o plano gerado, há na Imagem 1 o resultado de um problema com 3 anéis.

```
==>
43 move r1 r2 t1 t3 t3
68 move r2 r3 t1 t2 t2
115 move r1 t3 t3 r2 t2
127 move r3 t1 t1 t3 t3
135 move r1 r2 t2 t1 t1
199 move r2 t2 t2 r3 t3
238 move r1 t1 t1 r2 t3
root 1
1 shiftTower t1 t2 t3 -> m-shiftTower 6
6 selectDirection r1 t1 t2 t3 -> m-selectDirection 8
8 selectDirection r2 t1 t3 t2 -> m-selectDirection 288
42 move_abstract t1 t3 -> newMethod21 43
64 exchange t1 t3 t2 -> exchangeLR 65 289
65 move_abstract t1 t2 -> newMethod21 68
114 move_abstract t3 t2 -> newMethod21 115
123 exchange t3 t2 t1 -> exchangeRL 124 290
124 move_abstract t1 t3 -> newMethod21 127
134 move_abstract t2 t1 -> newMethod21 135
180 exchange t2 t1 t3 -> exchangeLR 198 291
198 move_abstract t2 t3 -> newMethod21 199
237 move_abstract t1 t3 -> newMethod21 238
286 exchange t1 t3 t2 -> exchangeClear
288 selectDirection r3 t1 t2 t3 -> selectedDirection 292
289 rotateTower t3 t2 t1 -> m-rotateTower 114 123
290 rotateTower t2 t1 t3 -> m-rotateTower 134 180
291 rotateTower t1 t3 t2 -> m-rotateTower 237 286
292 rotateTower t1 t3 t2 -> m-rotateTower 42 64
```

Imagem 1 - Plano para um problema em HDDL

## **Análise Crítica**

Após uma análise dos problemas e do domínio, ele parece trocar as torres em que os discos são empilhados usando uma task shiftTower. Essa task é a principal do domínio já que mediante chamadas recursivas (recursividade de cauda) dentro dessa task a pilha de anéis é transportada da torre t1 para a torre t3 utilizando uma torre intermediária t2. Outra coisa é que o predicado goal\_on não parece ser útil já que é utilizado para declarar a ordem da pilha dos anéis na torre objetivo. Isso, no entanto, já é estabelecido no goal.

## **Ferramentas**

Uma das ferramentas apresentadas no artigo, sendo uma ferramenta de transformação entre PDDL e HDDL, pode ser encontrada no link:

<https://github.com/ronwalf/HTN-Translation/tree/master>