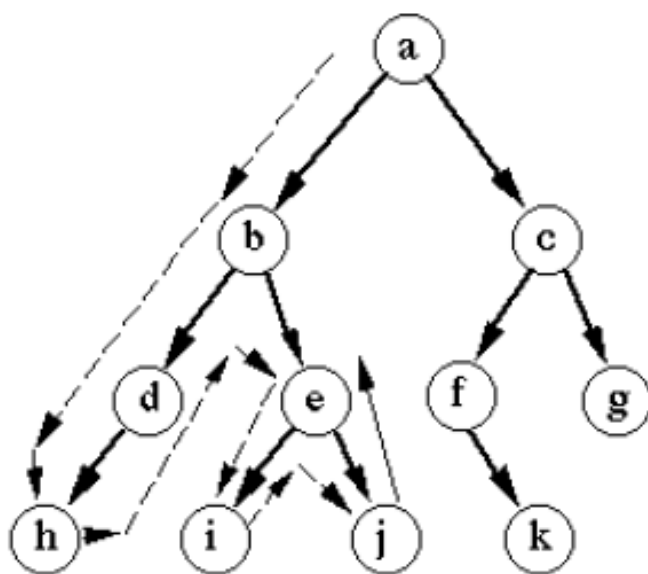


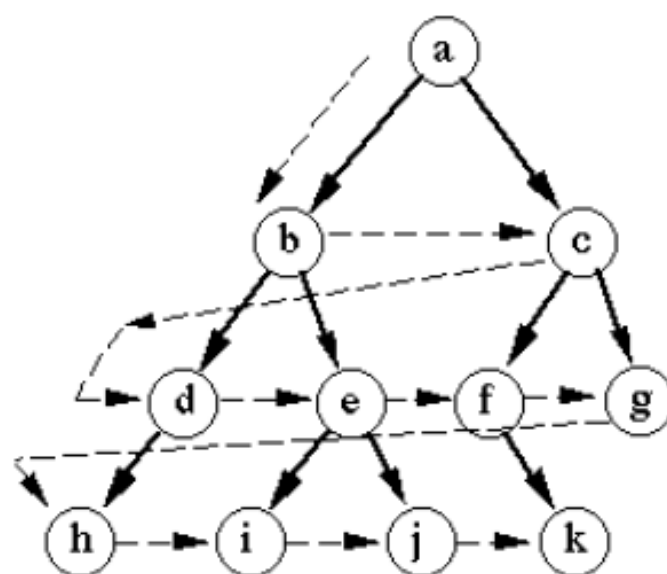
# Busca Informada (Heurística)

Professor: Igor da Penha Natal

# Recapitulação das aulas anteriores



Busca em Largura



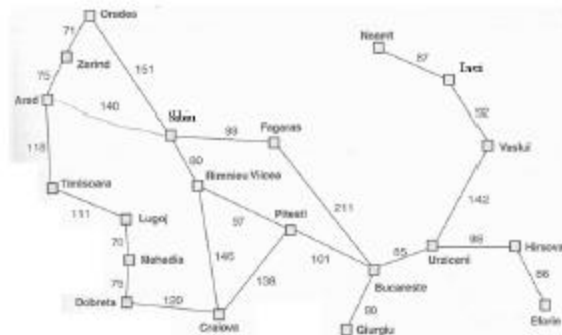
Busca em Profundidade

# Definição e exemplos de problemas de busca

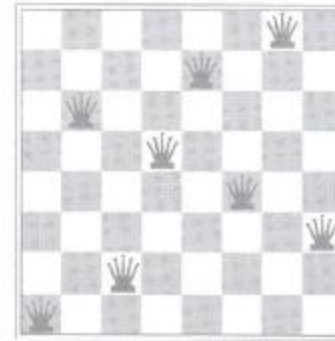
---

- Definição de um problema:
  - Estado inicial
  - Ações possíveis
    - Estado inicial + realização de ações possíveis = espaço de estados (espaço de busca)
  - Teste de objetivo: determina se um dado estado é um estado objetivo.
  - Função de custo: medida de desempenho da busca dentro do processo de resolução do problema

# Definição e exemplos de problemas de busca



A solução é o caminho entre uma cidade e outra.  
A busca retorna o caminho enquanto o encontra.



A solução é um posicionamento específico das 8 rainhas.  
A busca retorna o tabuleiro "solução".



Estado inicial



Estado objetivo

A solução é um posicionamento específico das pedras do tabuleiro.

A busca retorna o processo para encontrar a solução.

# Definição e exemplos de heurística

---

- A busca informada (heurística) utiliza conhecimento específico do problema para poder encontrar soluções de maneira mais eficiente.
- Tentativa de expandir os caminhos mais **promissores** primeiro.
- Heurística auxilia a encontrar os nós mais promissores a cada passo.
  - Heurística é a função que **estima** a distância ao objetivo.

# Definição e exemplos de heurística

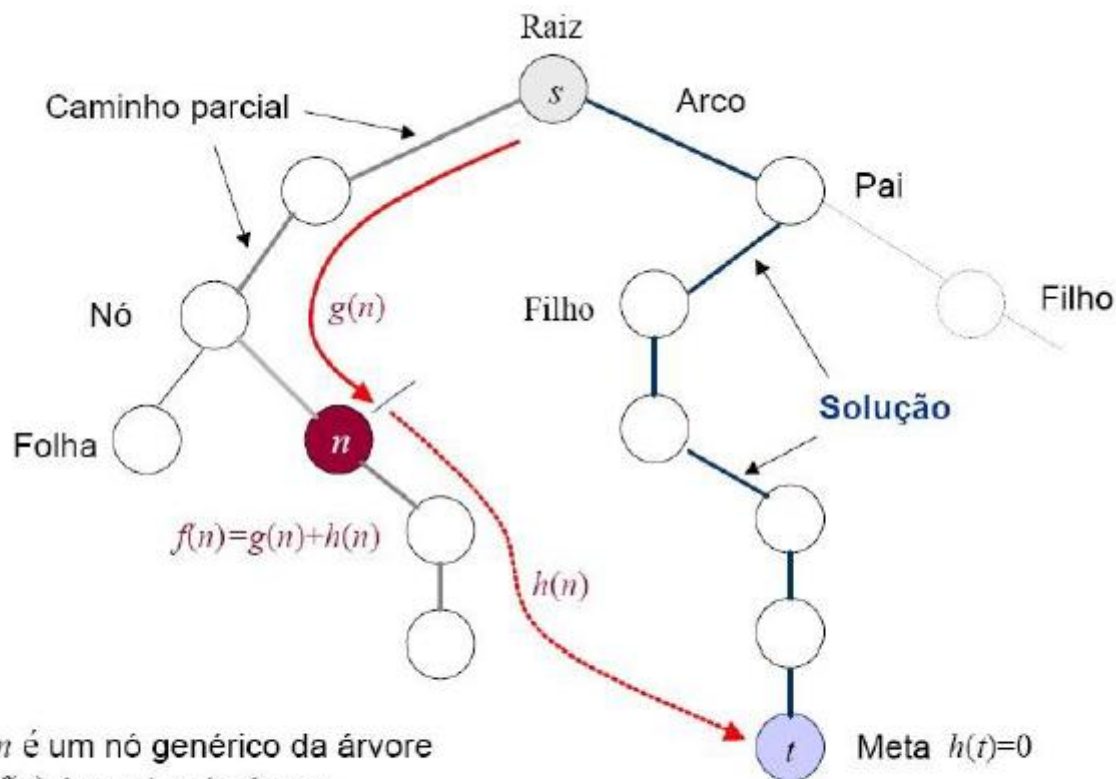
- Abordagem Geral: A ideia é usar uma função de avaliação  $f(n)$  para cada nó, onde:
  - Será estimado o grau em que um nó é “desejável” como caminho.
  - Serão expandido os nós mais desejáveis.

$$f(n) = g(n) + h(n)$$

$g(n)$  = Custo do caminho do estado inicial até o nó  $n$

$h(n)$  = Custo estimado de  $n$  ao **estado objetivo** pelo caminho mais barato

# Definição e exemplos de heurística



$n$  é um nó genérico da árvore

$f(n)$  é o valor de  $f$  em  $n$

$g(n)$  é o custo da trajetória até  $n$

$h(n)$  é uma *estimativa* do custo de  $n$  até a meta

# Definição e exemplos de heurística

---

- Busca de Custo Uniforme:  $f(n) = g(n)$ .
- Busca pelo melhor primeiro (Gulosa):  $f(n) = h(n)$ .
- Busca  $A^*$ :  $f(n) = g(n) + h(n)$ .



# Definição e exemplos de heurística

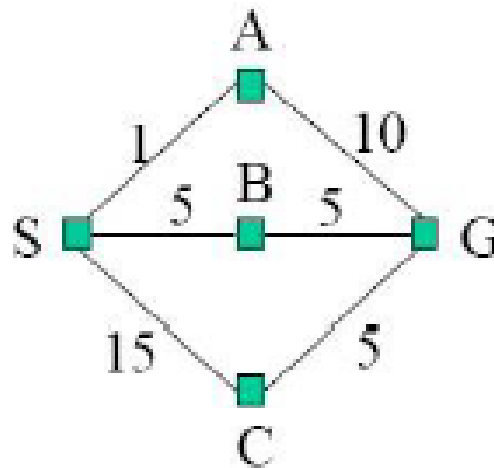
---

- Definição de heurísticas admissíveis:
  - Mesmo sem conhecer o custo real, muitas vezes é possível garantir que o custo estimado não o ultrapassa.
- Exemplos:
  - Quantidades de números fora de posição no quebra-cabeça de 8: para cada número fora de posição, é necessário ao menos um movimento para coloca o número na posição correta.
  - Rota mínima entre duas cidade: qualquer caminho entre duas cidade é maior ou igual ao comprimento da reta que une as duas cidades.

# Algoritmos de Busca Heurística

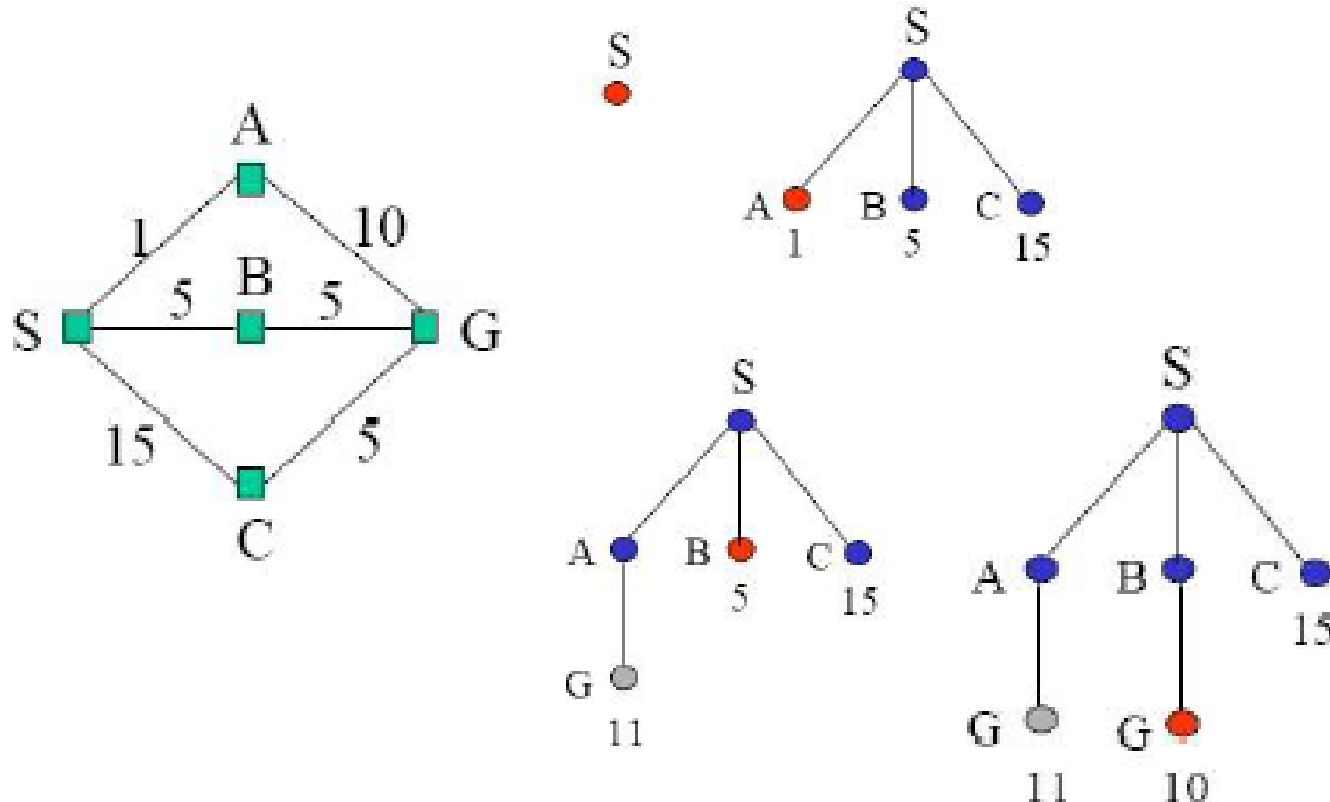
# Algoritmo de custo uniforme

- Leva em conta apenas o fator de  $g(n)$  a cada passo.
- Considere o problema de encontrar o menor caminho entre os nós S e G no grafo a seguir:



# Algoritmo de custo uniforme

- Considere o problema de encontrar o menor caminho entre os nós S e G no grafo a seguir:



# Algoritmo busca pelo melhor primeiro (gulosa)

---

- Leva em conta apenas o fator heurístico  $h(n)$ .
- Expande o nó mais próximo à meta, na suposição de que isso provavelmente levará a uma solução rápida e eficiente.

# Algoritmo busca pelo melhor primeiro (gulosa)

---

- Pensando no problema do Puzzle a seguir:
  - Configuração Inicial:

2	8	3
1	6	4
7		5

- Configuração Objetivo:

1	2	3
8		4
7	6	5

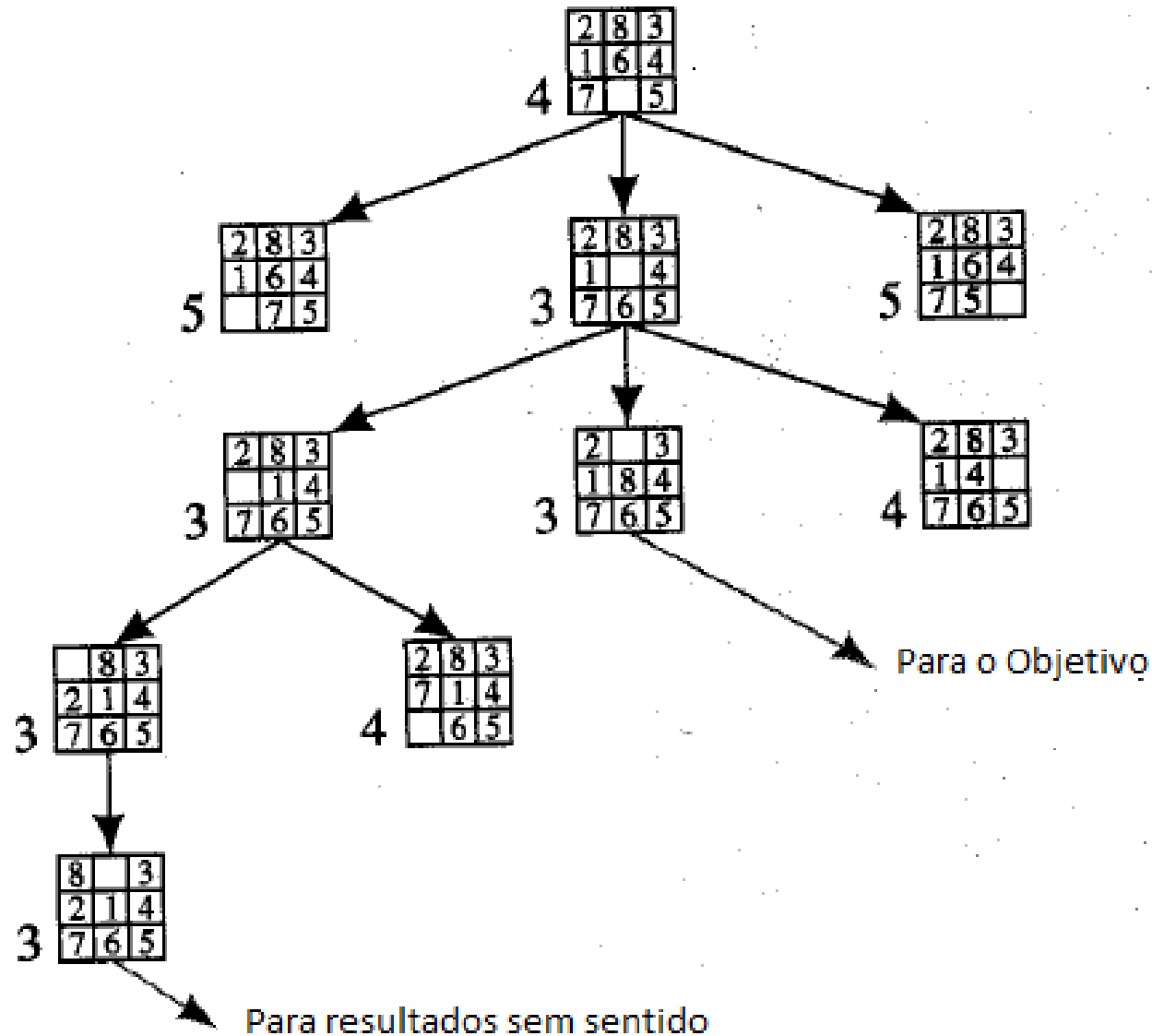
- Heurística: Quantidade de números fora de posição.

# Algoritmo busca pelo melhor primeiro (gulosa)

---

	2	8	3
	1	6	4
4	7		5

# Algoritmo busca pelo melhor primeiro (gulosa)





# Algoritmo A\*

---

- Está técnica requer que a estimação do custo  $f(n) = g(n) + h(n)$  seja monotônica, ou seja, nunca caia quando se afasta da raiz da árvore de busca. Diferente das duas técnicas anteriores, sob esta hipótese, sempre é possível encontrar a solução ótima com a busca A\*.
- A busca A\* é completa, ótima e eficiente.
- No entanto sua complexidade ainda é exponencial e o seu uso de memória é intenso.
- Ele guarda todos os nós para expandir o que têm menor valor de  $f(n)$ .

# Algoritmo A\*

$$f(n) = g(n) + h(n)$$

$h(n)$  otimista  $\rightarrow$  heurística admissível

$f(n)$  nunca decresce ao longo de um caminho  $\rightarrow$  monotonicidade

$n$  pai de  $n'$ ,  $f(n') \leq f(n) \rightarrow f(n') = \max[f(n), g(n') + h(n')] \rightarrow \text{pathmax}$

A\* expande todos nós com  $f(n) < f^*$

Primeira solução encontrada é a solução ótima

A\* é otimamente eficiente  $\rightarrow$  nenhum outro algoritmo expande menos nós que A\*

# Algoritmo A\*

---

- Pensando no problema do Puzzle a seguir:

- Configuração Inicial:

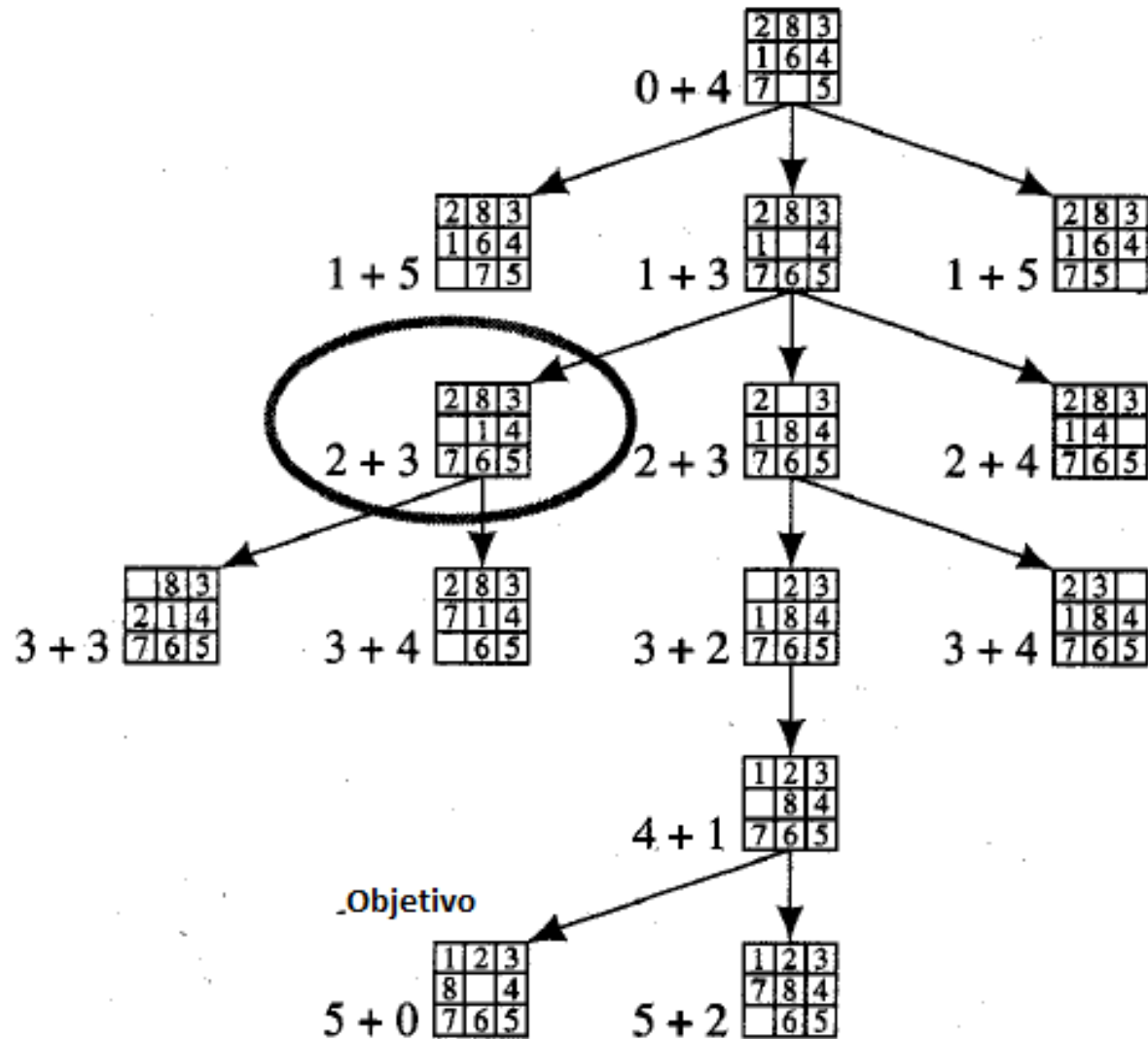
2	8	3
1	6	4
7		5

- Configuração Objetivo:

1	2	3
8		4
7	6	5

- Heurística: Quantidade de números fora de posição.

# Algoritmo A\*



# Síntese da Aula

---

- Nesta aula vimos
  - Busca Informada (Heurística)
    - Definição de problema de busca e exemplos.
    - Definição e exemplos de heurísticas.
    - Algoritmos
      - Custo Uniforme;
      - Guloso;
      - A\*.

# Próxima Aula

---

- Outras Buscas
  - Busca Local;
  - Simulated Annealing;
  - Busca Tabu;
  - Hill Climbing;
  - Algoritmo Genético.

# Referências Bibliográficas

---

- [1] RUSSEL, S.; NORVIG, P.; Artificial Intelligence – A modern approach. 4ª edição. Editora: Pearson Education Limited, 2021.
- [2] POOLE, D.; MACKWORTH; Artificial Intelligence: Foundations of Computational Agents. 2ª edição. Editora: Cambridge University Press, 2017.