Christophe Rodrigues

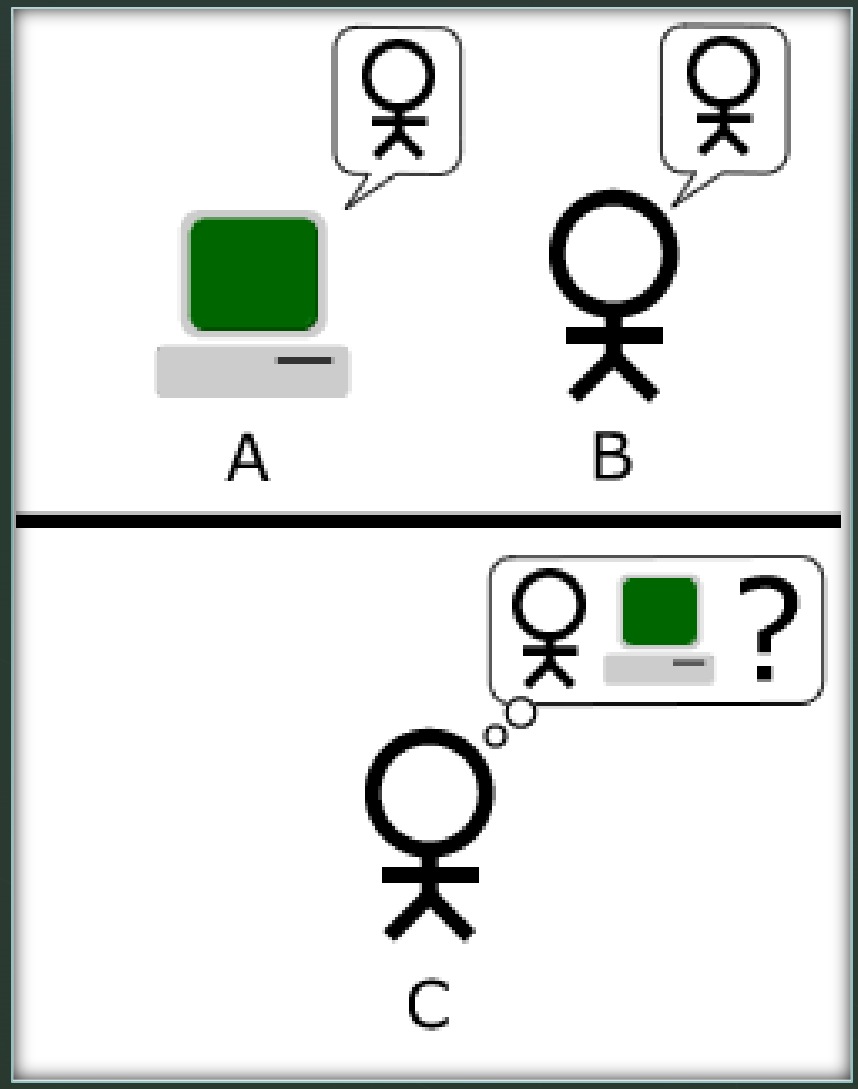
DataScience & IA

Plan du cours

- Les débuts de l'IA
- Règles d'associations
- A*

Début de l'IA

- Reflexion de Turing (1950) : Comment évaluer si un ordinateur ou programme a un comportement intelligent?
- Il propose le fameux test de Turing



Eliza - Premier agent conversationnel

- Weizenbaum (1964) système à base règles (ou grammaire)
- Agent psychiatre

```
Human:  Men are all alike.
ELIZA: In what way?
Human:  They're always bugging us about something.
ELIZA: Can you think of a special example?
Human:  Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here?
Human:  He says I'm depressed much of the time.
ELIZA: I'm sorry to hear that you are depressed much of the time.
Human:  It's true.  I'm unhappy.
ELIZA: Do you think coming here will help you not to be unhappy?
Human:  I need some help, that much seems certain.
```

Plus récemment...

- L'agent conversationnel Tay développé par Microsoft (2016)
- L'agent était disponible en ligne sur tweeter et tout le monde pouvait intéragir avec l'IA. Cet agent était doté de mécanisme d'apprentissage et apprenait de ces intéractions avec les utilisateurs.
- Très vite (<24heures), les utilisateurs ont orientés les réponses de l'agent...

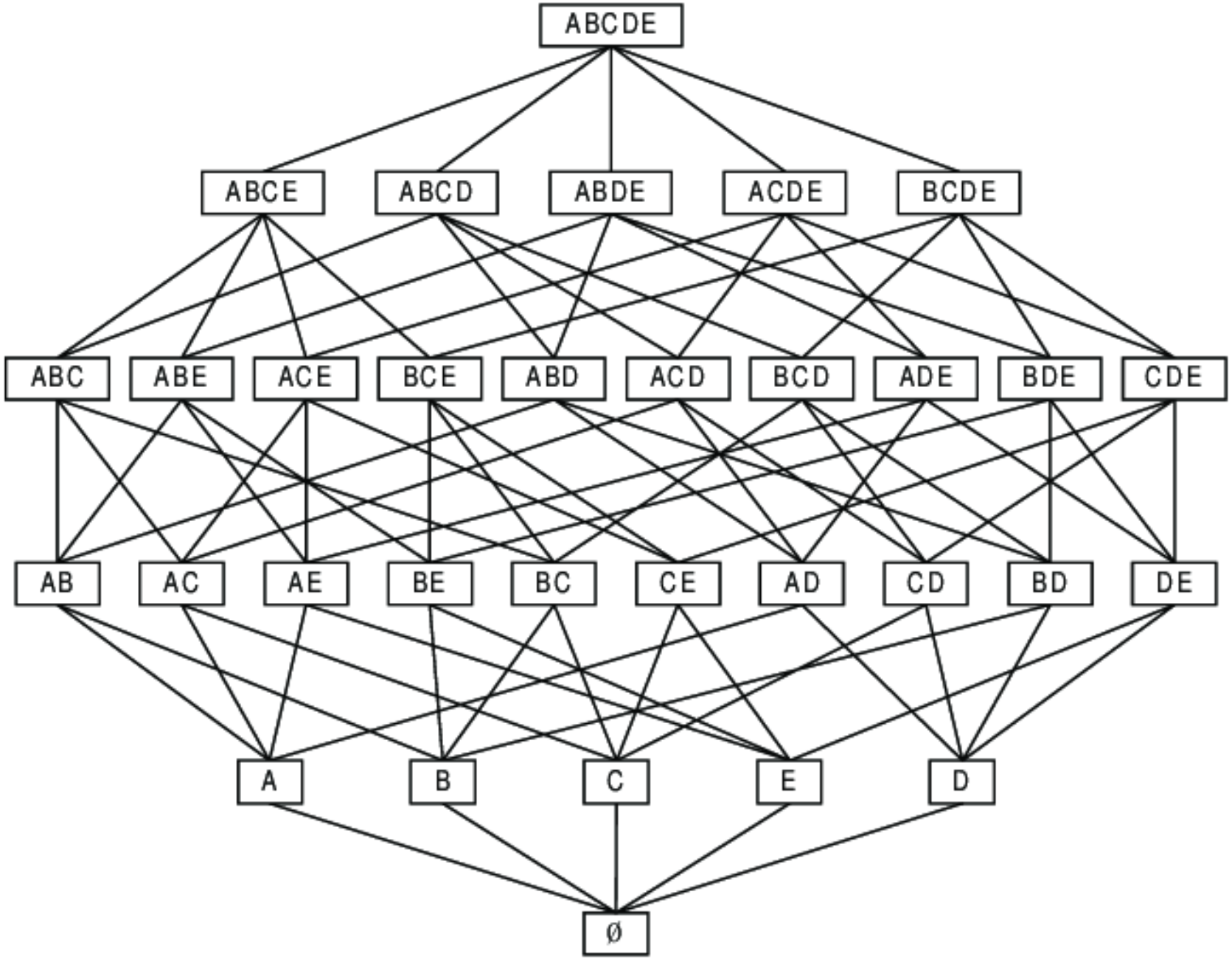
Fouille de données - Règles d'associations
DataMining - Association rule learning

- Basket analysis problem – APriori (Agrawal, 1994)
- But: trouver quels produits sont achetés ensemble afin:
  - optimiser les promotions
  - optimiser l'agencement des rayons

Contexte

- Ticket de caisse = ensemble d'articles
- Transaction = ensemble d'items

- But : découvrir les associations d'items fréquents (au dessus d'un certain seuil)

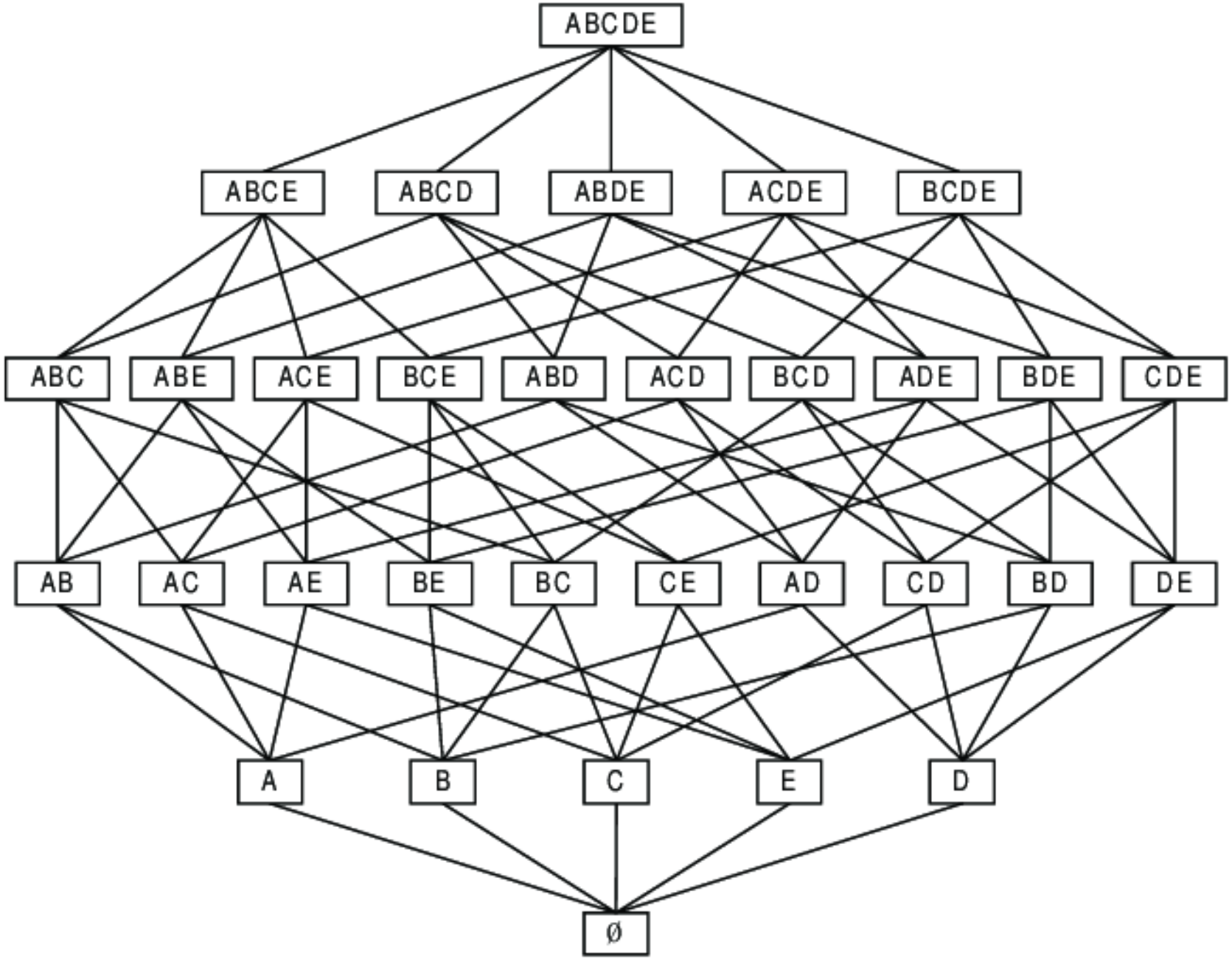

ABCDE
ABCE
ABCD
ABDE
ACDE
BCDE
ABC
ABE
ACE
BCE
ABD
ACD
BCD
ADE
BDE
CDE
AB
AC
AE
BE
BC
CE
AD
CD
BD
DE
A
B
C
E
D
Ø

Taille de l'espace de recherche

- Dans un hypermarché moyen : 40000 références possibles
- => 2^40000 patterns possibles
- (pour info, il y a approximativement un peu plus de 2^25 étoiles dans l'univers)

Apriori

- Parcours en largeur des régles d'associations
- Commence par les règles constituées d'un item
  - Puis deux, trois,…, jusqu'à la taille de l'alphabet (tous les différents items)
- Mécanisme d'élagage exploitant la structure de l'espace de recherche
  - Monotonicité : un sur-ensemble ne peut être plus fréquent qu'un ensemble

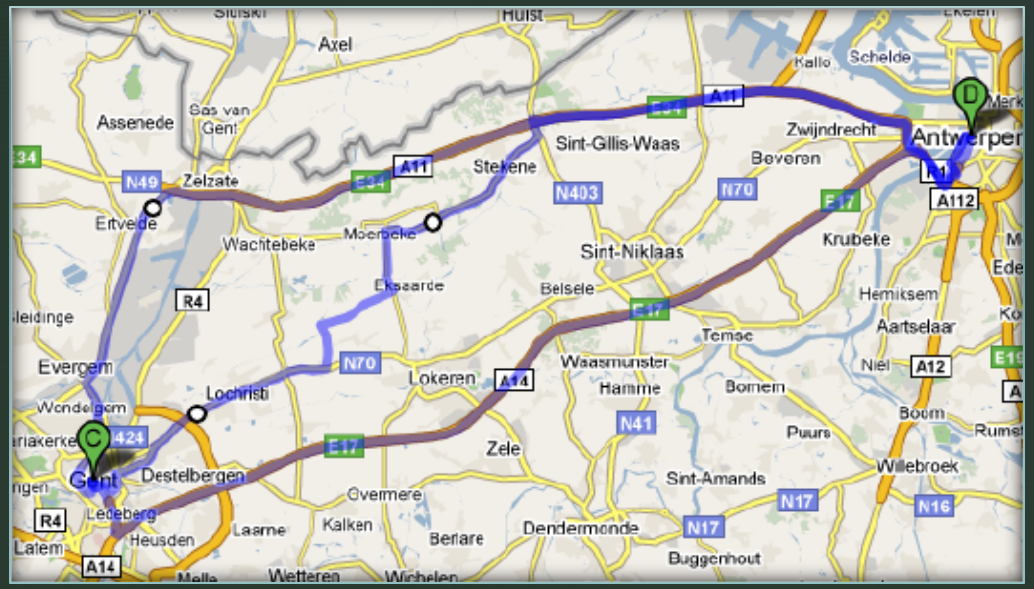

ABCDE
ABCE
ABCD
ABDE
ACDE
BCDE
ABC
ABE
ACE
BCE
ABD
ACD
BCD
ADE
BDE
CDE
AB
AC
AE
BE
BC
CE
AD
CD
BD
DE
A
B
C
E
D
Ø

Pour aller plus loin

- APriori était le première solution proposée pour la découverte de règles d'associations.
- Depuis, beaucoup d'autres méthodes et de variantes ont été proposés.
- Le framework libre SPMF propose actuellement plus de 169 méthodes et capable de traiter des transactions et des règles de différentes sortes:
  - http://www.philippe-fournier-viger.com/spmf/

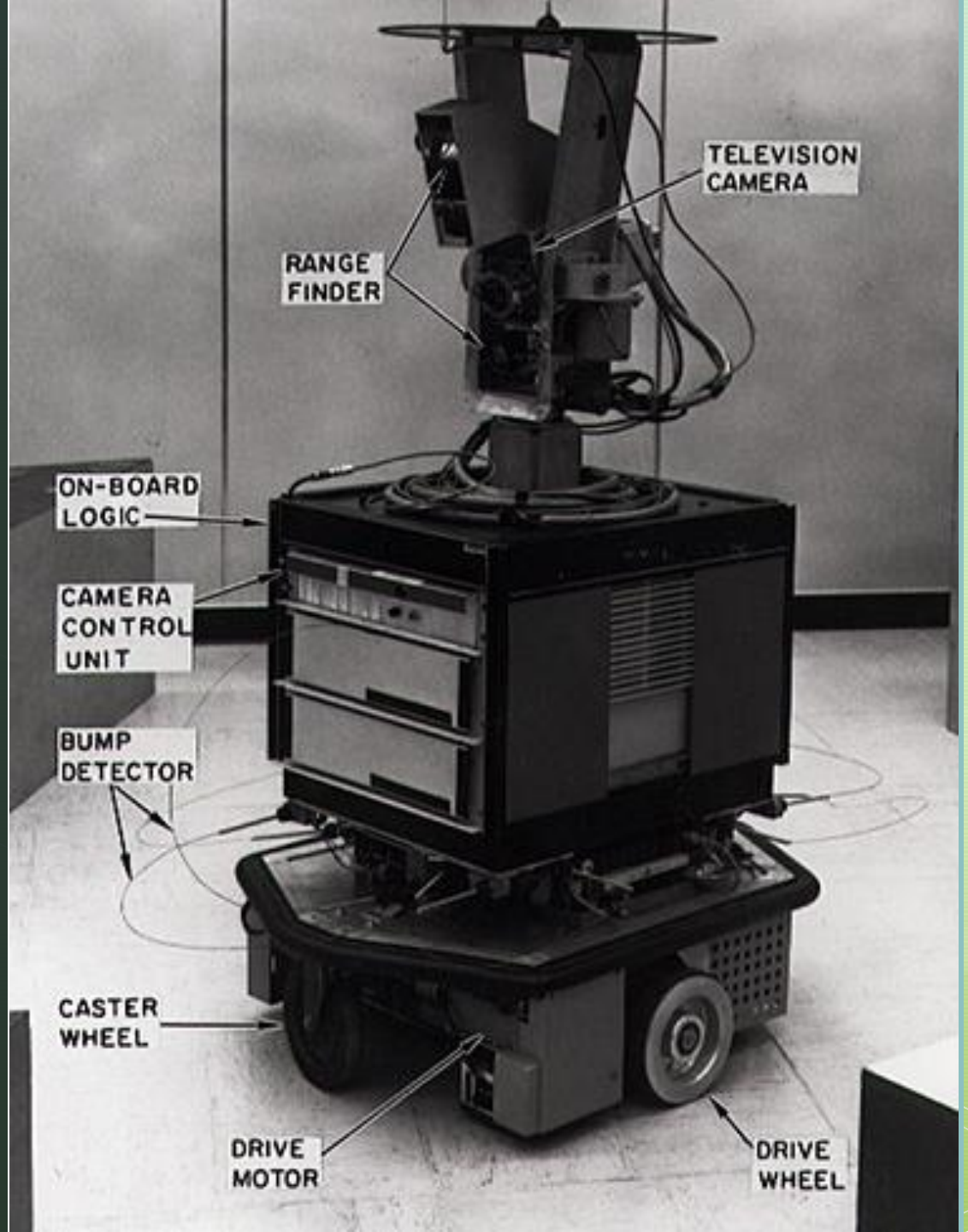
A*
recherche d'un chemin - Pathfinding

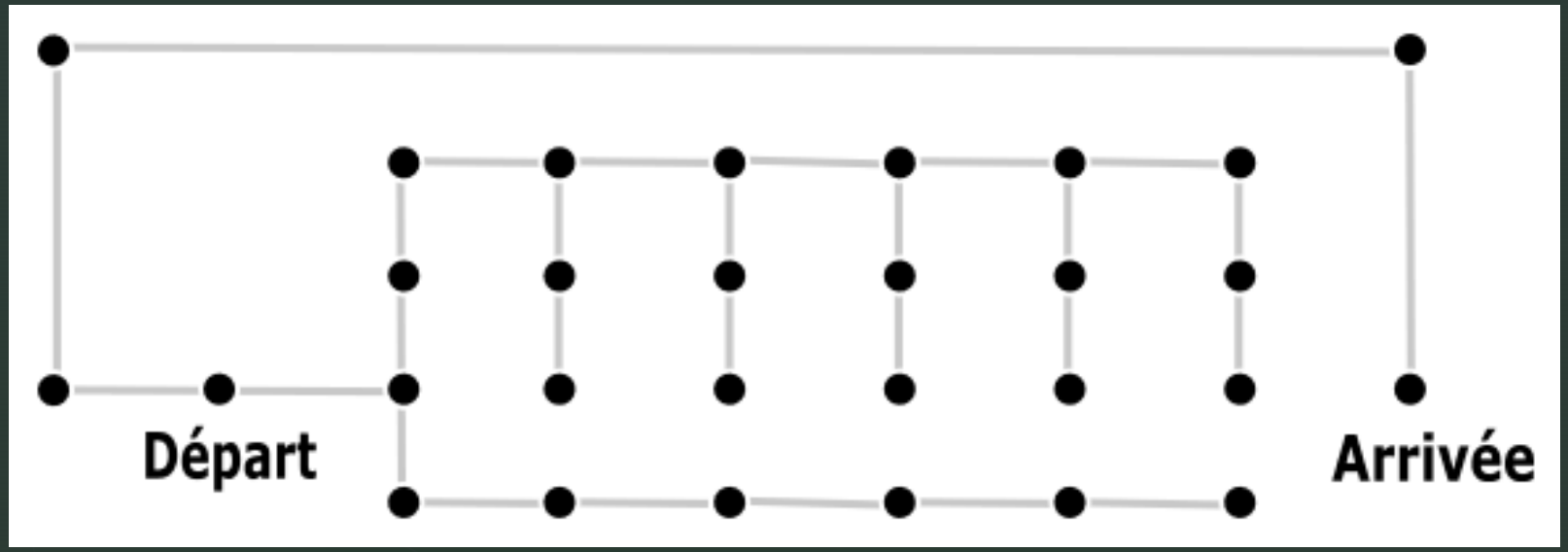
- Problème: Comment trouver un chemin entre deux points parmis tous les chemins possibles



A*

- Proposé par Hart, Nilsson et Raphael en 1968
- Contexte : améliorer les déplacements du robot Shakey
- Contrainte : trouver une bonne solution le plus vite possible (non nécessairement la meilleure de toutes)





- Méthode la plus utilisée pour trouver le plus court chemin:
  - L'algorithme de Dijkstra (1959) va parcourir le graphe en largeur "en aveugle" et finir par trouver le plus court de tous les chemins entre le nœud de départ et le nœud d'arriver.
- A* va être guidé par une information supplémentaire (heuristique) afin de s'orienter dans le graphe. L'heuristique la plus commune est la distance au but à vol d'oiseau. Cela fonctionne très bien sur les cartes/graphes classiques mais pas toujours...

Évaluation de l'intérêt d'un noeud

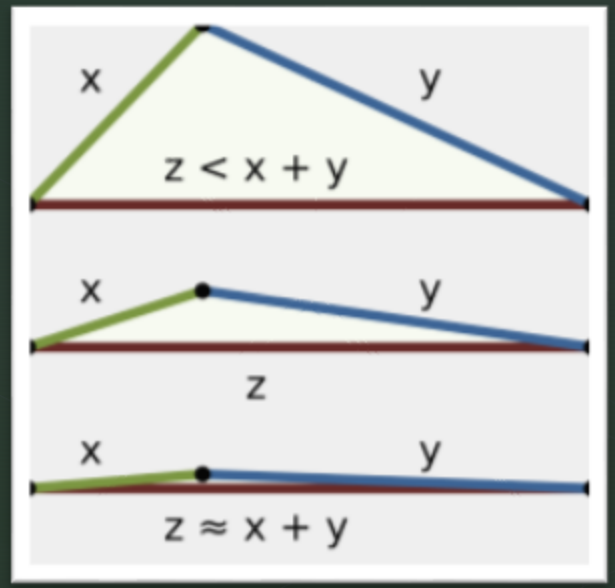
- Fonction de coût f au noeud n est décomposée en deux parties:
- f(n) = g(n) + h(n)
- Avec g(n) le coût cumulé entre le noeud de départ et le noeud courant n : "le passé"
- Avec h(n) le coût estimé du noeud courant n au noeud de destination : "le futur estimé"

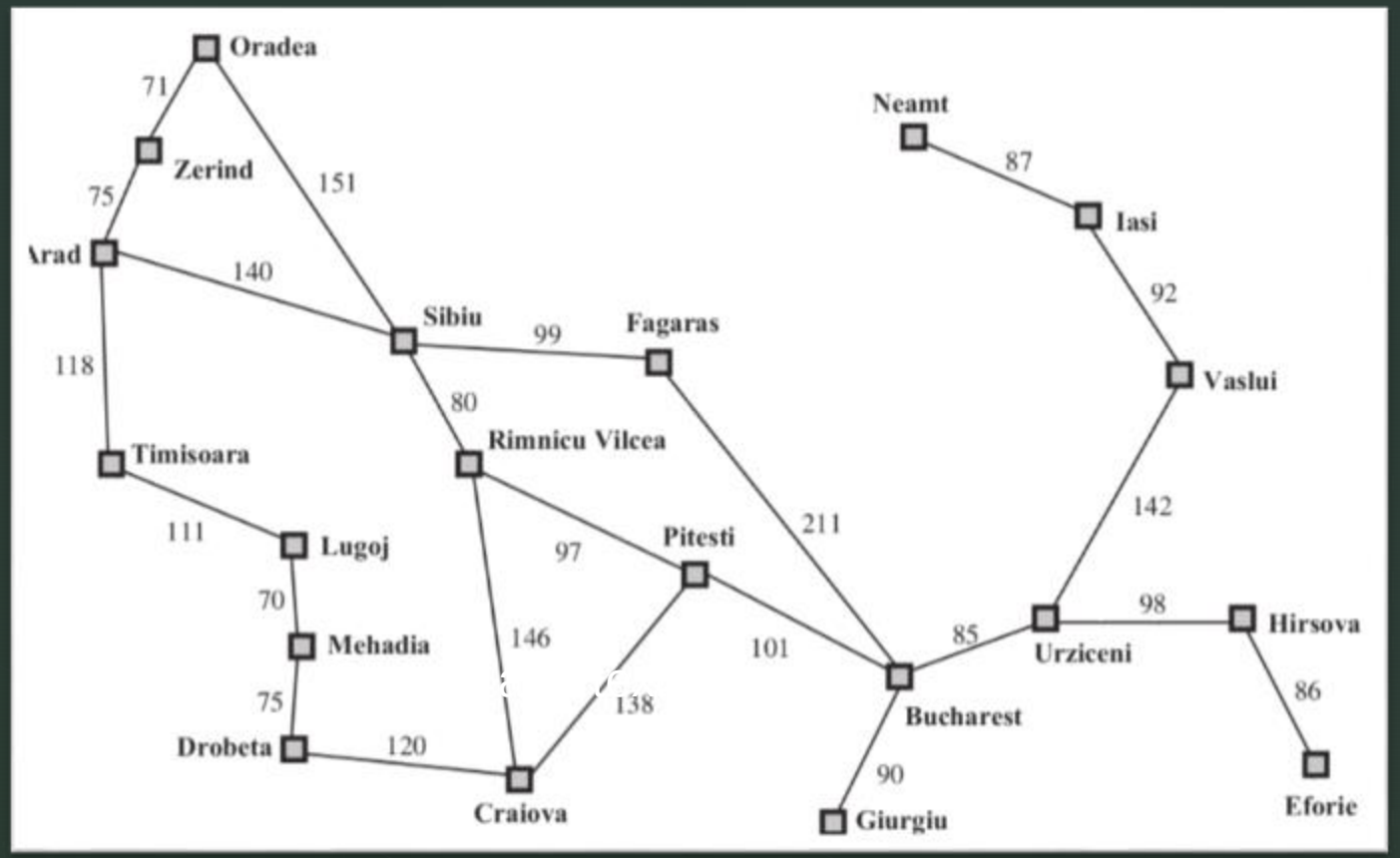
Condition pour l'optimalité

- Il est possible de garantir l'optimalité d'une heuristique pour un problème donné.
- Pour être optimale l'heuristique doit être:
  - admissible
  - monotone

Condition pour l'optimalité

- Admissibilité
  - Une heuristique est admissible si elle ne surestime jamais le coût pour atteindre le but (optimiste).
- Monotonicité
  - $h(n) \leq c(n,n') + h(n')$
  - Forme d'inégalité triangulaire
- Exemple: distance à vol d'oiseau

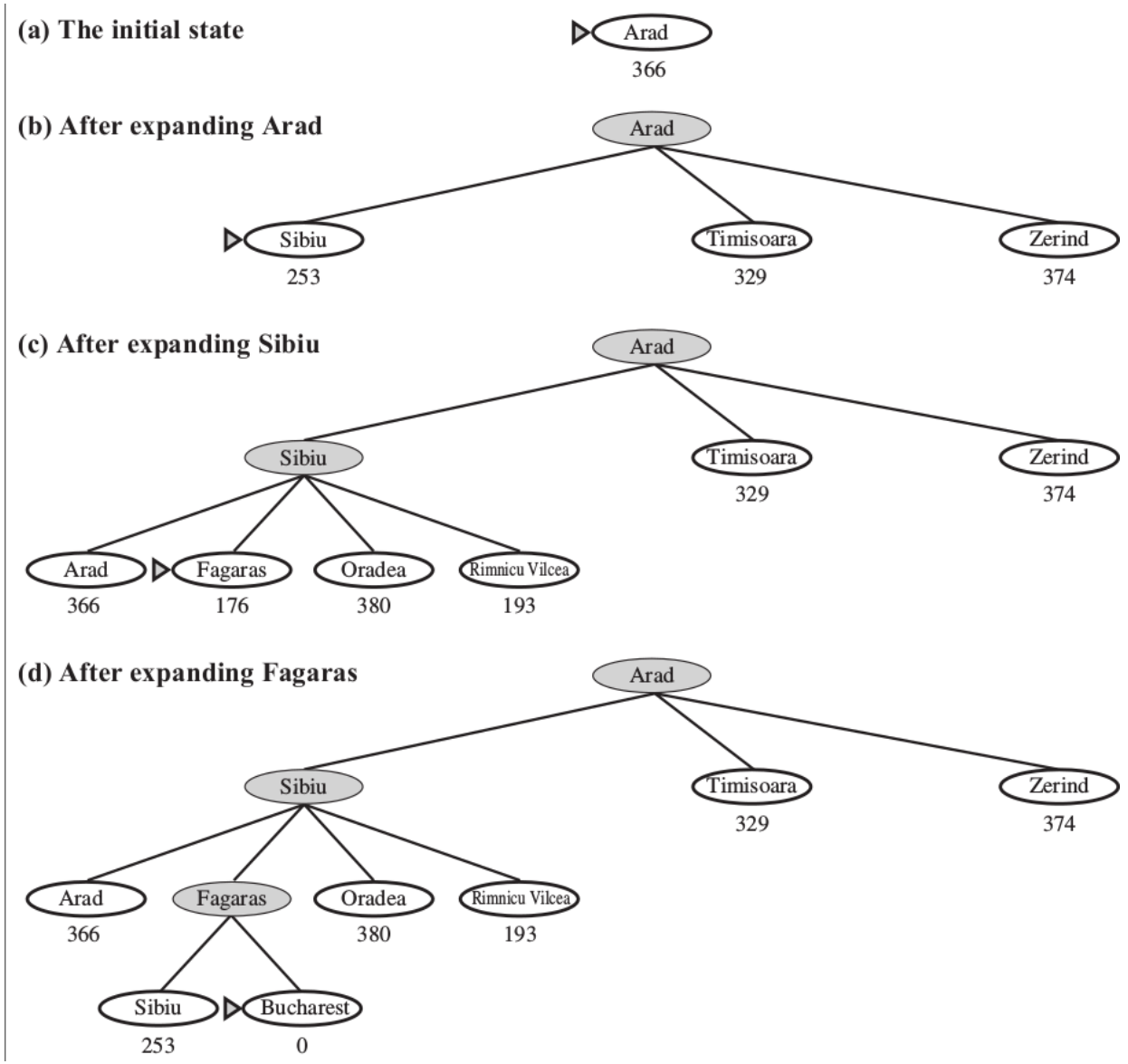




Noeud de départ : Arad

Noeud d'arrivée : Bucharest

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374


(a) The initial state
Arad
366
(b) After expanding Arad
Arad
Sibiu
253
Timisoara
329
Zerind
374
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
176
Oradea
380
Rimnicu Vilcea
193
(d) After expanding Fagaras
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
Oradea
380
Rimnicu Vilcea
193
Sibiu
253
Bucharest
0