

Proseminar Künstliche Intelligenz

Teil 1: Allgemeine und heuristische Suchverfahren

Vorgetragen von:

- Henning Eberhardt:
 - Allgemeine Suchverfahren
 - Constraint Satisfaction Problems
- Clemens Lode:
 - Heuristische Suchverfahren

1. Teil: Uninformiertes Suchen

- Allgemeine Suchalgorithmen
 - Breitensuche
 - Tiefensuche
 - Probleme
- CSP – Constraint Satisfaction Problems
 - Backtracking
 - Forward tracking

Bewertungskriterien

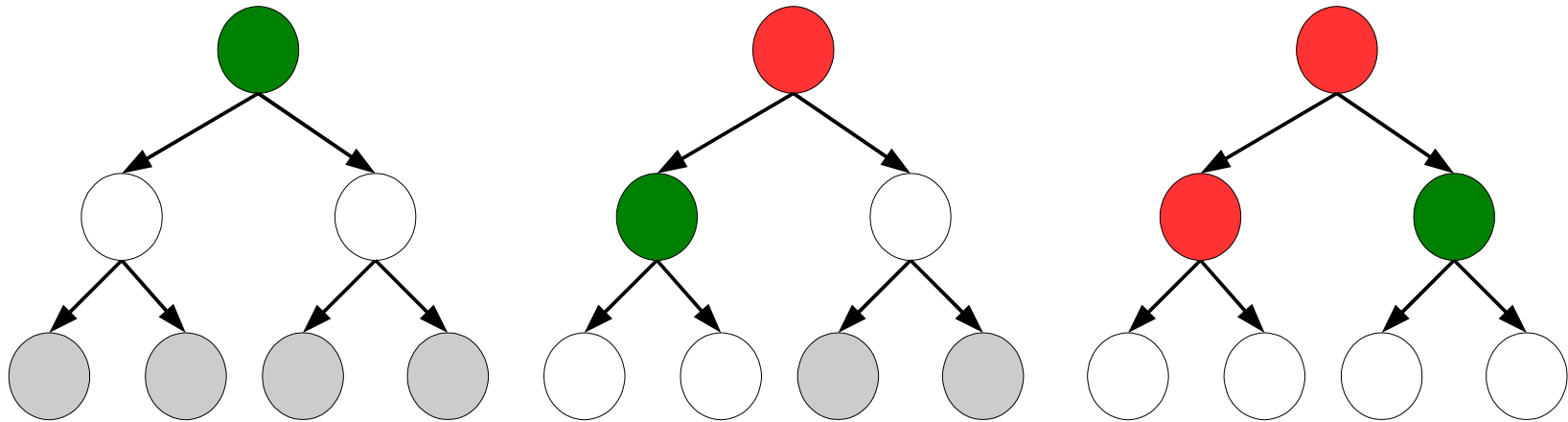
- Vollständigkeit
- Optimalität
- Zeitkomplexität
- Speicherbedarf

Komplexität

- Verzweigungsfaktor(**branchingfactor**): **b**
- Suchtiefe(**depth**): **d**
- Maximale Knotenzahl:

$$1 + b + b^2 + b^3 + \dots + b^d = \sum_{i=0}^d b^i$$

Breitensuche (Breadth-first search)



Breitensuche (Breadth-first search)

- Vollständig: Ja
- Optimal: Ja
- Zeitaufwand: $O(b^{d+1})$
- Speicheraufwand: $O(b^{d+1})$

Breitensuche (Breadth-first search)

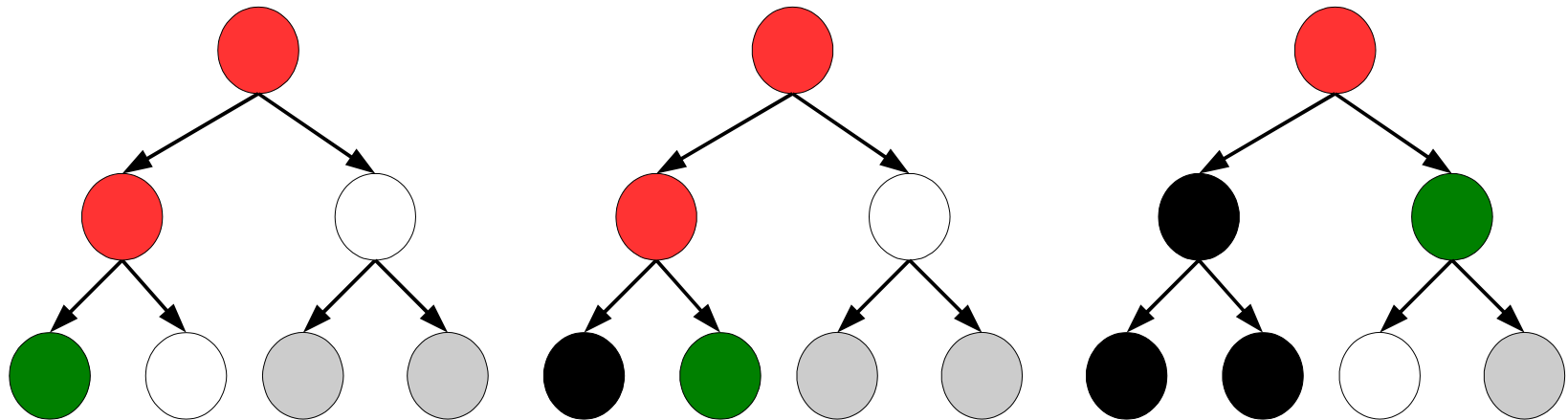
Tiefe	Knoten	Dauer	Speicherverbrauch
2	1100	0.11s	1Mb
4	111100	11s	106Mb
6	10^7	19m	10Gb
8	10^9	31h	1Tb
10	10^{11}	129d	101Tb
12	10^{13}	35y	10P(eta)b
14	10^{15}	3523y	1E(xa)b

Uniform-Cost Search

- Vollständig: Ja, sofern Pfadkosten > 0
- Optimal: Ja, sofern Pfadkosten > 0
- Zeitaufwand: $O(b^{\text{top}(C^* / e)})$
- Speicheraufwand: $O(b^{\text{top}(C^* / e)})$

C^* = Knotenzahl auf dem Lösungspfad

Tiefensuche (Depth-first search)



Tiefensuche

(Depth-first search)

- Vollständig: Nur bei endlichem Suchbaum
- Optimal: Nein
- Zeitaufwand: $O(b^{d+1})$
- Speicheraufwand: $O(b \cdot m)$

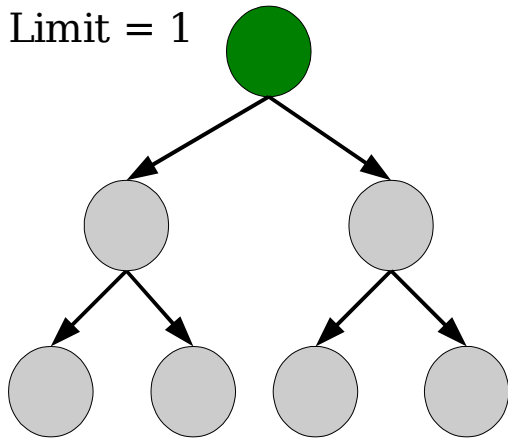
m sei maximale Suchtiefe

Begrenzte Tiefensuche (Depth-limited search)

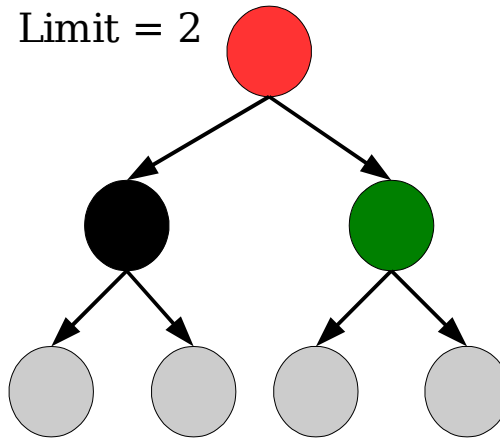
- Vollständig: Nur bei endlichem Suchbaum
- Optimal: Nein
- Zeitaufwand: $O(b^{d+1})$
- Speicheraufwand: $O(b \cdot m)$

Iterative deepening depth-first search

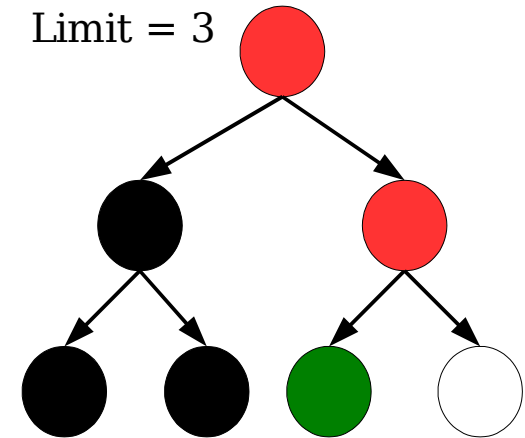
Limit = 1



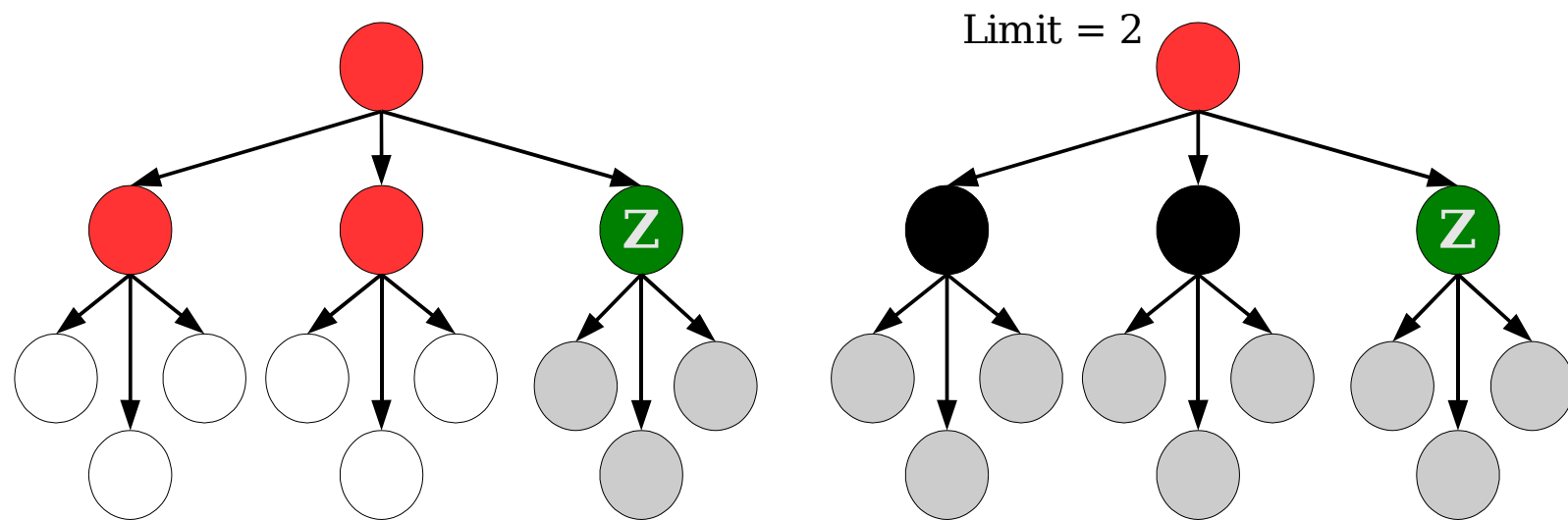
Limit = 2



Limit = 3



Iterative deepening depth-first search



- IDS ist billiger als Breitensuche
- IDS ist der bevorzugte Suchalgorithmus

Iterative deepening depth-first search

- Vollständig: Ja
- Optimal: Ja
- Speicheraufwand: $O(b \cdot d)$
- Zeitaufwand: $O(b^d)$

Bidirektionale Suche (Bidirectional search)

- Vollständig: Ja
- Optimal: Ja
- Zeitaufwand: $O(b^{d/2})$
- Speicheraufwand: $O(b^{d/2})$
- Nicht immer anwendbar

Schleifen unterbinden

- Schleifen können Probleme unlösbar machen



- Vergleiche aktuellen Knoten mit Abgearbeiteten
z.B. durch Hashtabelle



- Linearer Speicheraufwand bei der Tiefensuche
geht verloren

Constraint Satisfaction Problems (CSP)

- CSP besteht aus:
 - Menge von Variablen
 - Menge von Anforderungen/Abhängigkeiten
- Suchbaumtiefe ist bekannt
- Formen der Tiefensuche sind zur Lösung beliebt

Backtracking search

Abb. a

a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

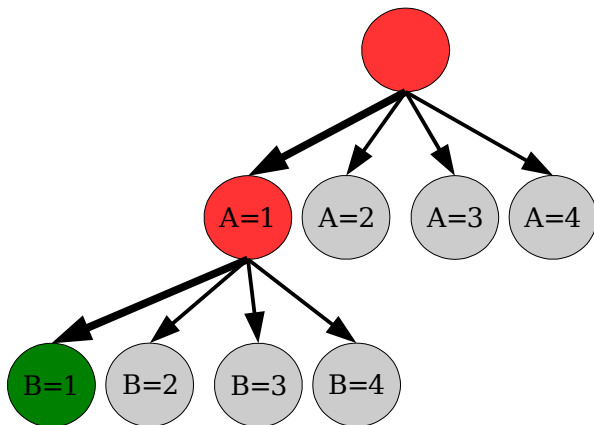


Abb. b

a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

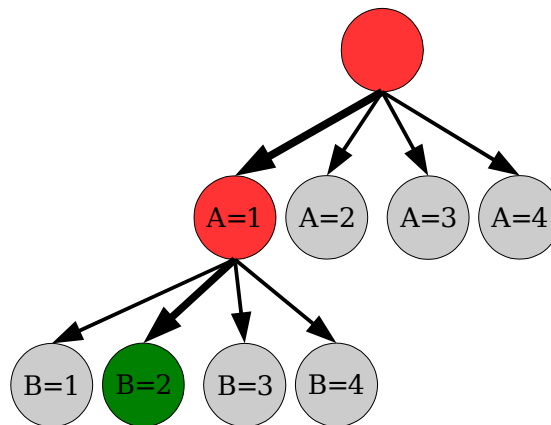
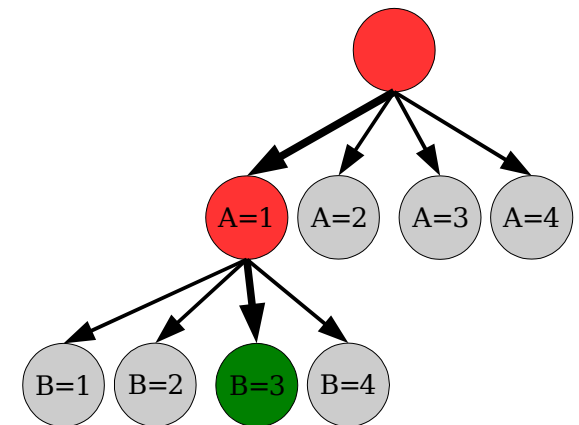


Abb. c

a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

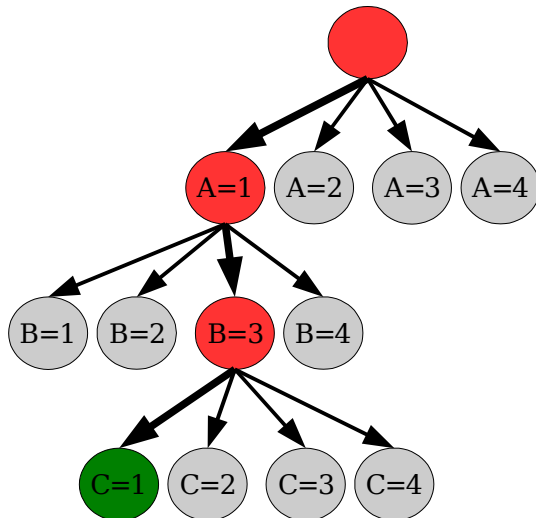


Backtracking search

Abb. d			
a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

...

Abb. d			
a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4



Forward checking

Abb. a

a	b	c	d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Abb. b

a	b	c	d
1			
2		2	2
3	3		3
4	4	4	

Abb. c

a	b	c	d
1			
2		2	2
3	3		3
4	4	4	

Forward checking

Abb. d

a	b	c	d
1			
2		2	
3	3		3
4	4		

Abb. e

a	b	c	d
1		1	1
2			
3		3	3
4	4		4

Abb. f

a	b	c	d
1		1	1
2			
3			3
4	4		4

Forward checking

Abb. g

	a	b	c	d
1	1		1	
2	2			
3	3			3
4	4	4		4