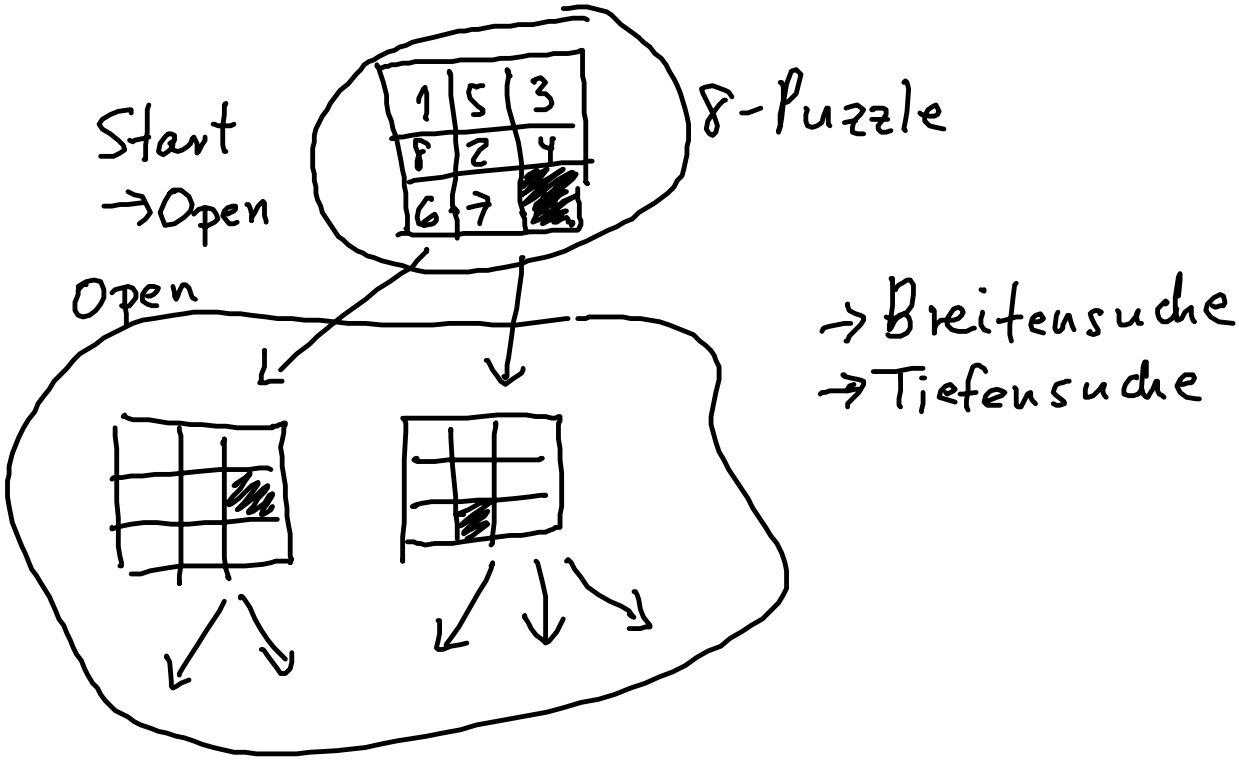
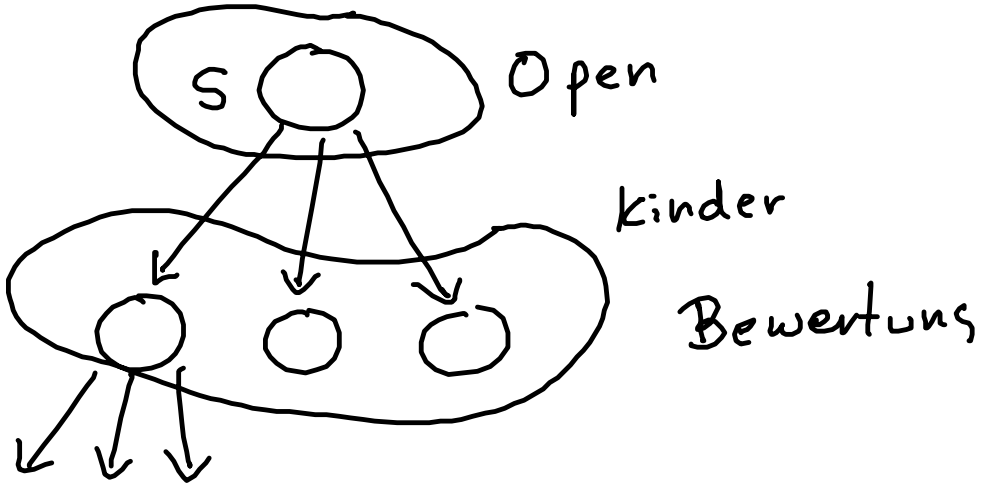
Zustandsraum



Bestensuche

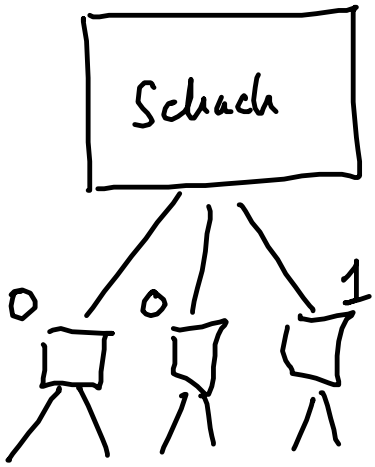


New Open := (Open − S) ∪ kinder

sortiert

„Map"
„Abstandsmaß"

→ Heuristik



Bewertungsfunktion $\in (0, \frac{1}{2}, 1)$

Bewertung

1 + 1 + 0
+ 1 + 0 + 1
+ 1 + 1 + 1

7 Fehler

Heuristik-1

2	1	3
6	5	7
8		4

1	2	3
4	5	6
7	8	

Heuristik-2

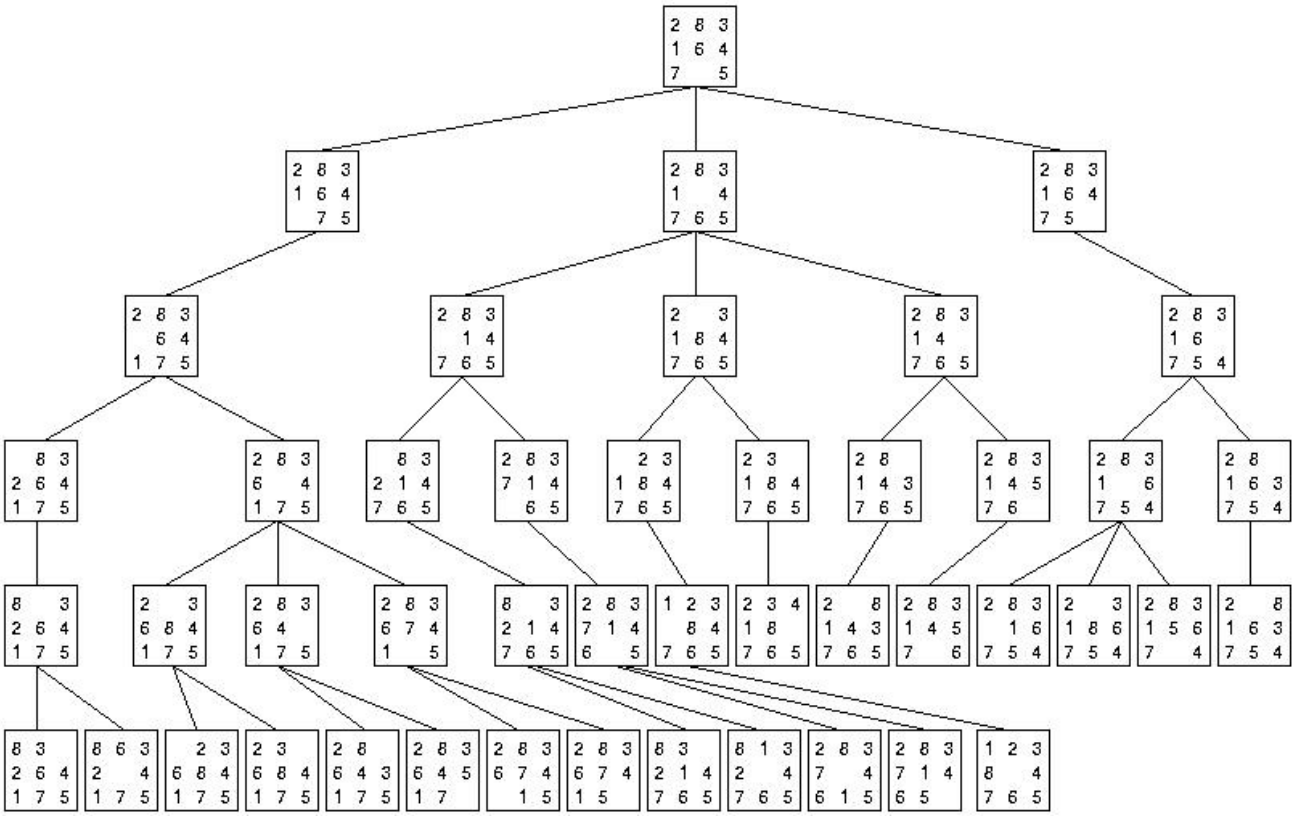
1 + 1 + 0
+ 2 + 0 + 3
+ 1 + 3 + 1

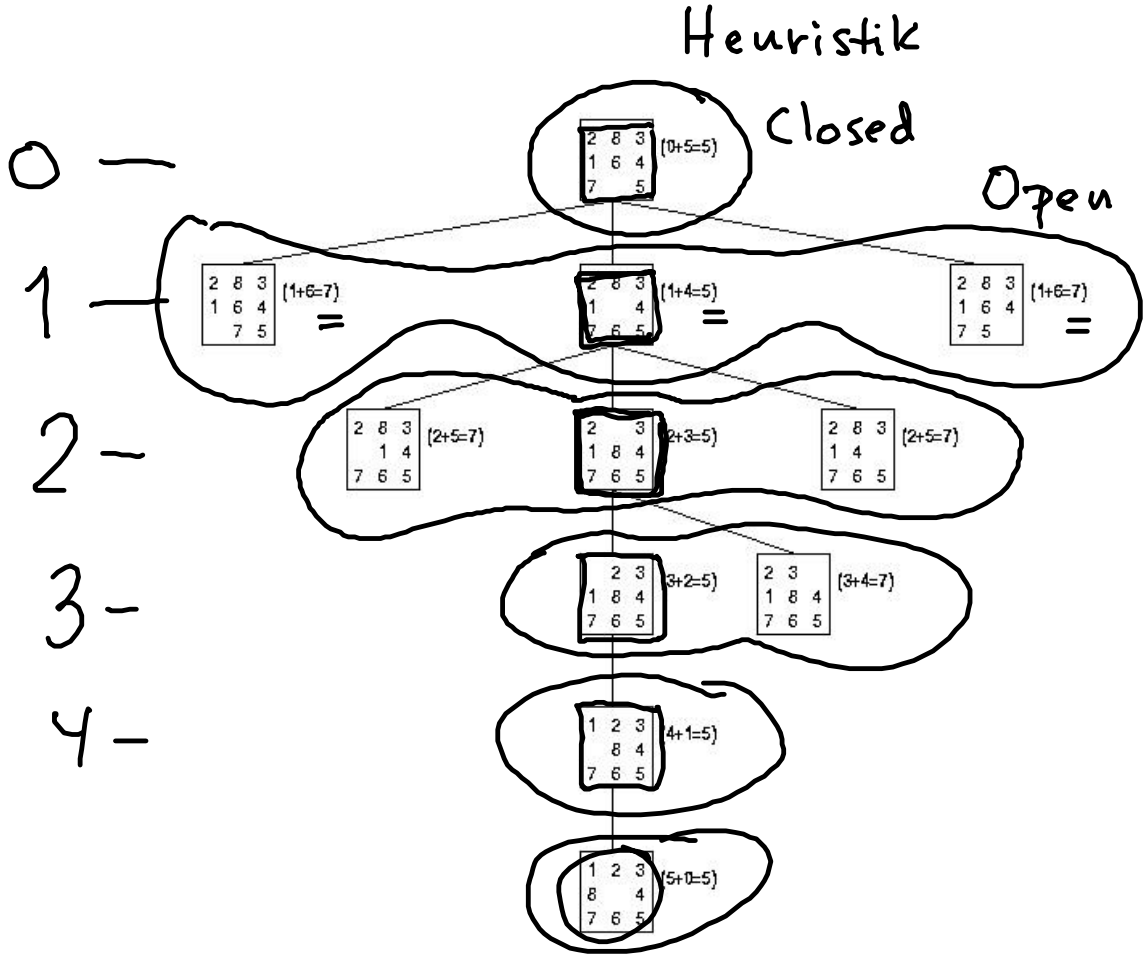
12

2 Fehler

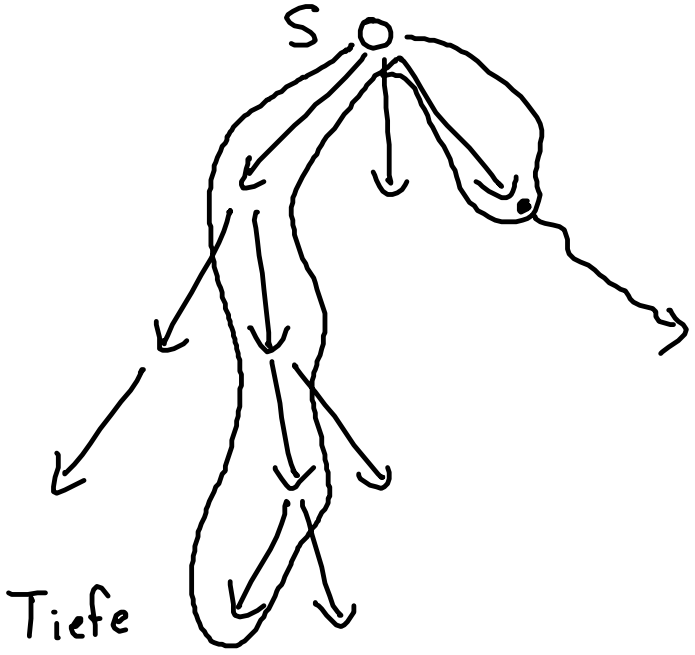
2	1	3
4	5	6
7	8	

Bestensuche



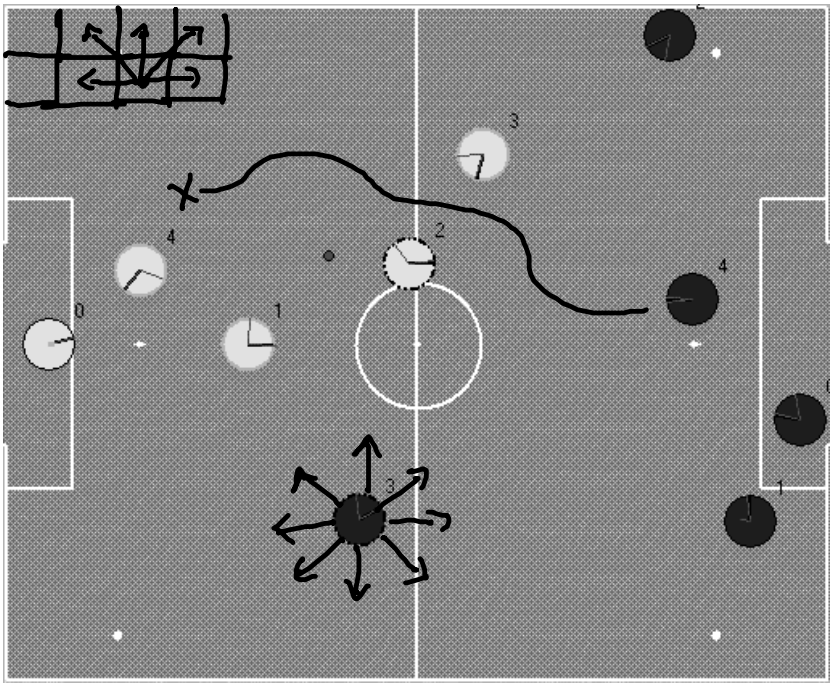


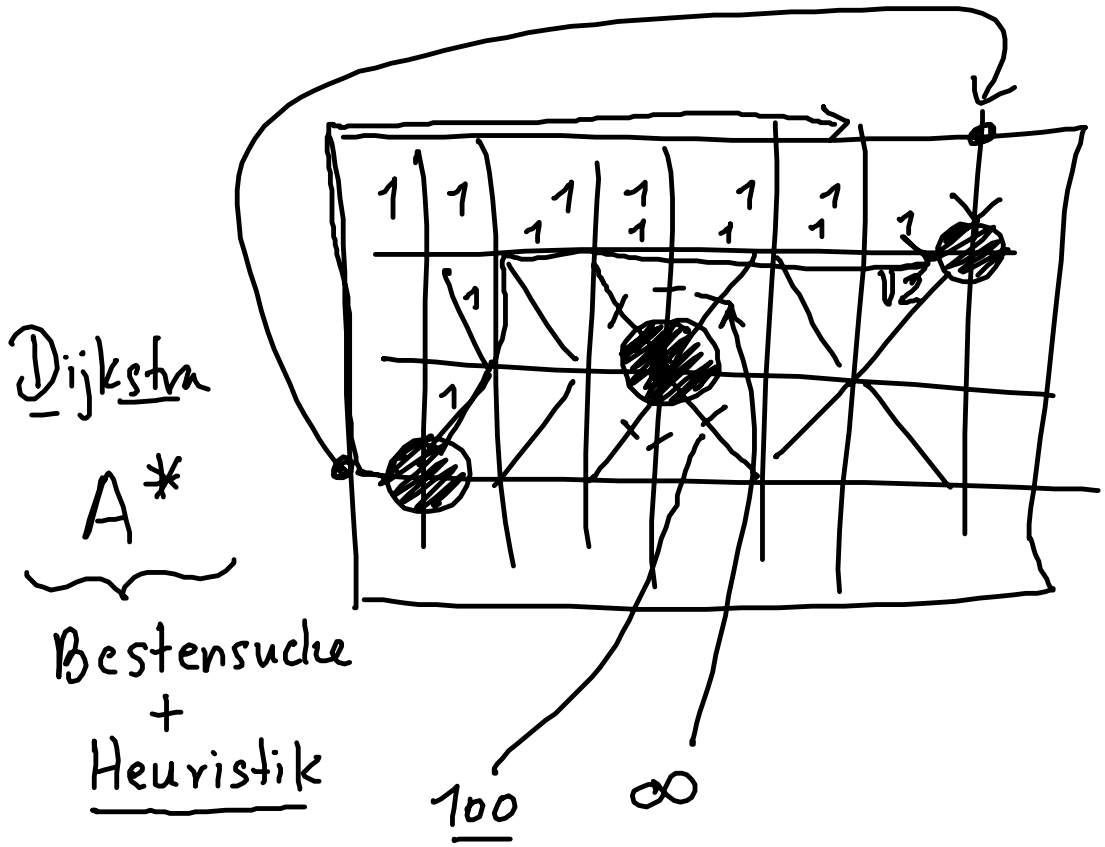
$$f(n) = g(n) + h(n)$$

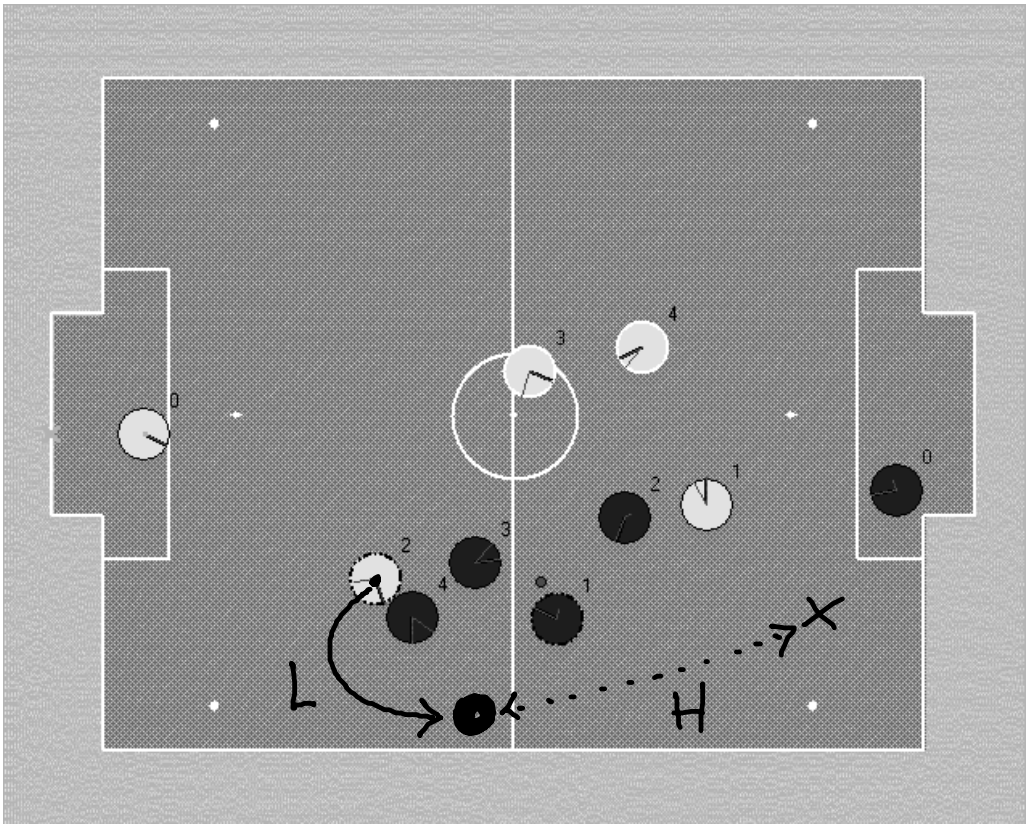


$$f(n) = g(n) + h(n)$$

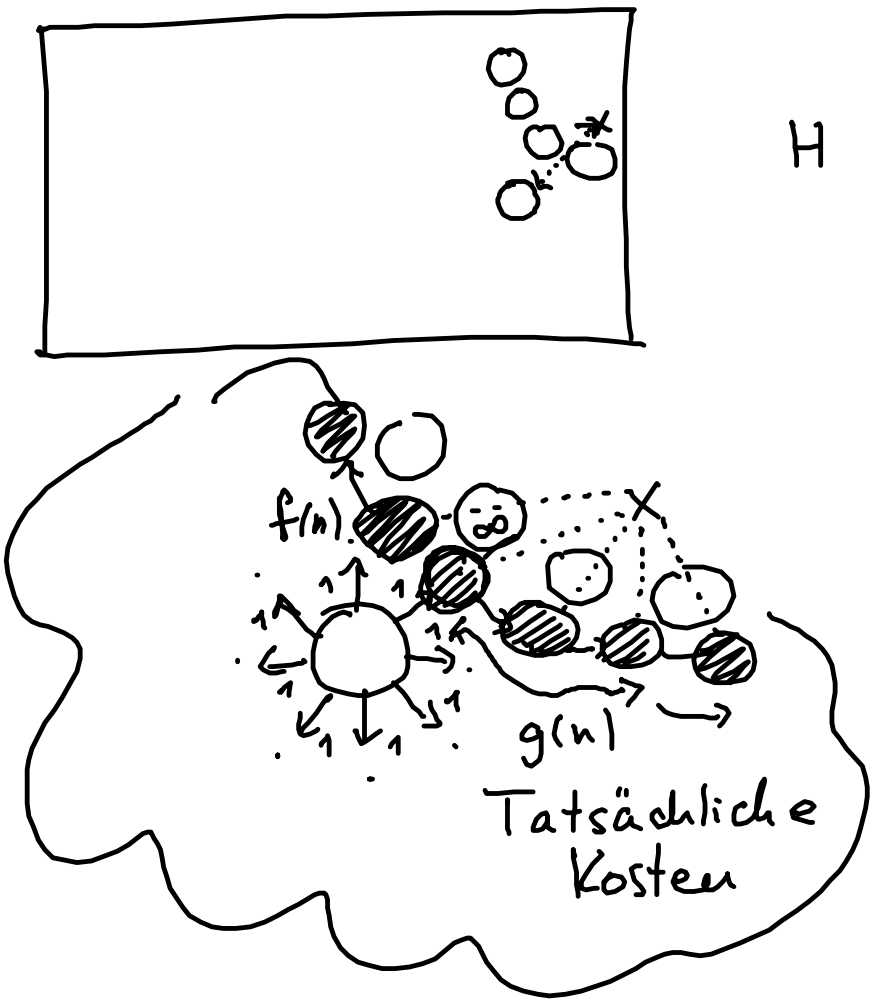
zu groß

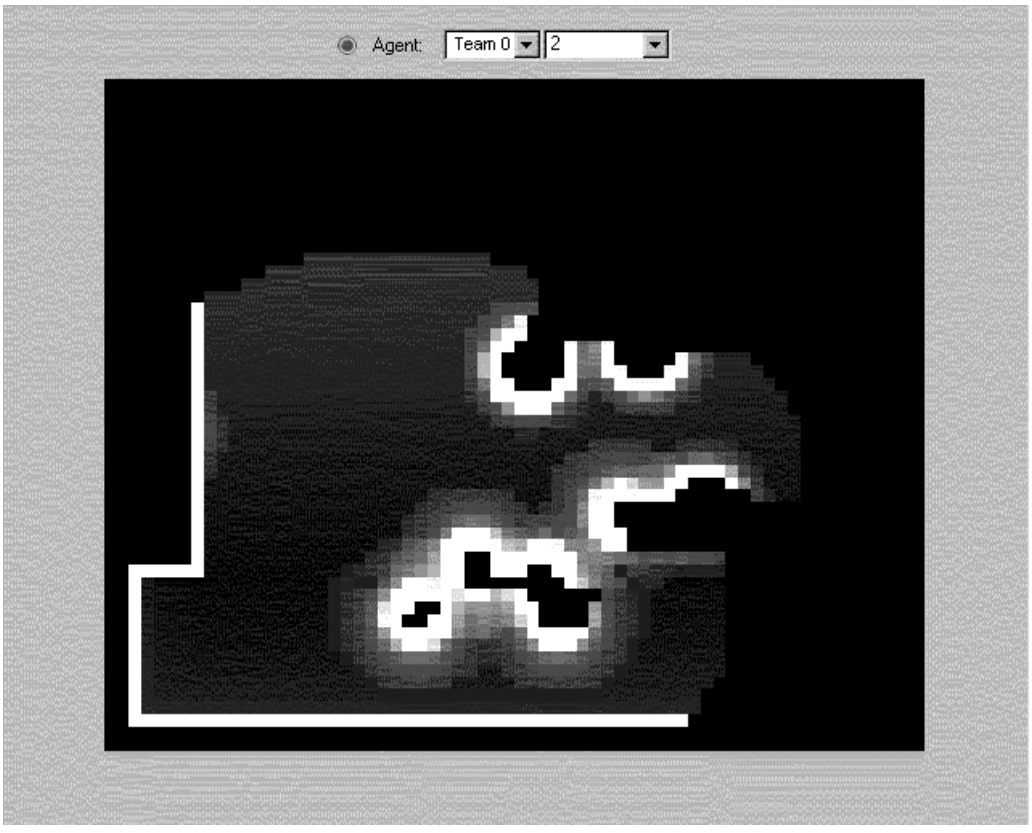


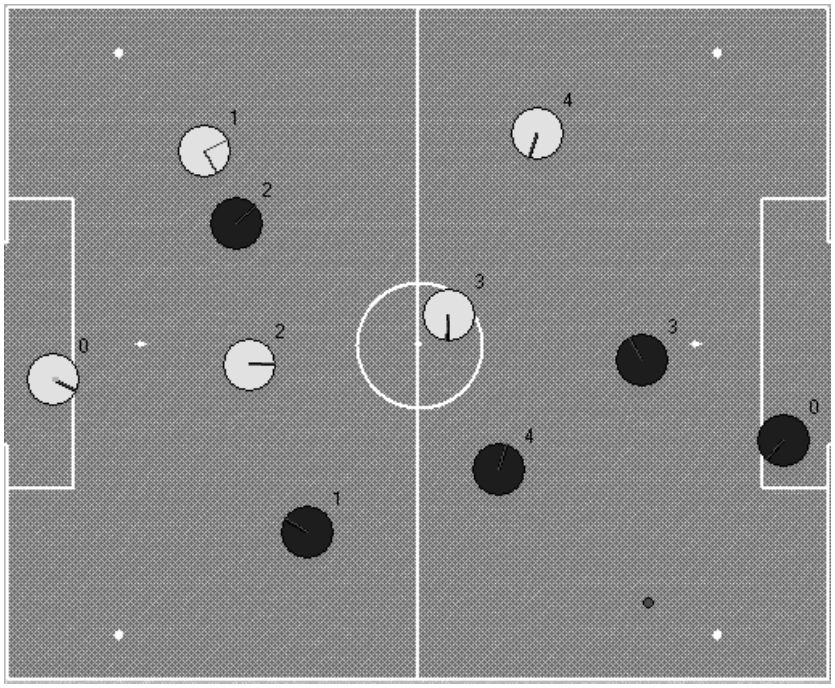


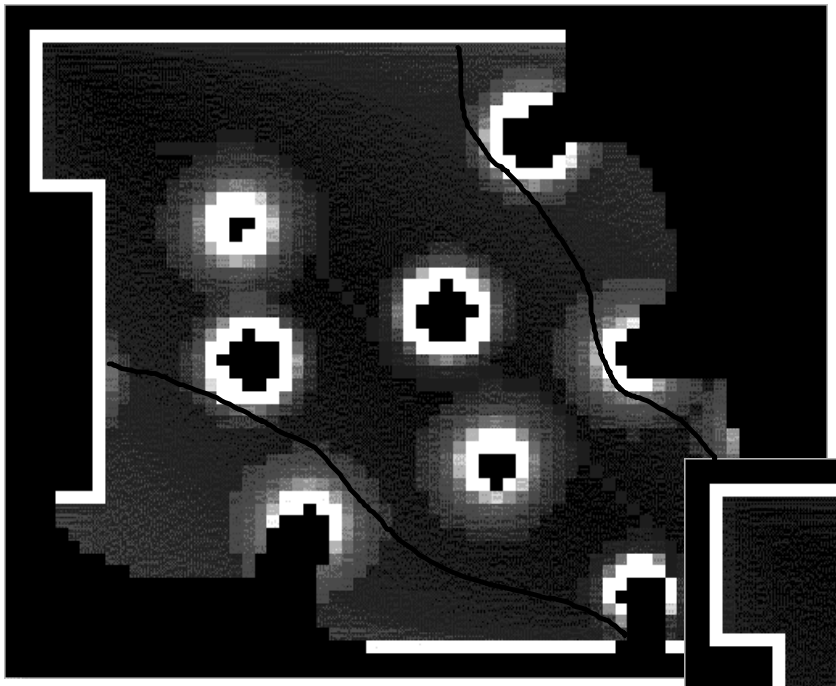


$$f(n) = \underset{L}{g(n)} + \underset{H}{h(n)}$$

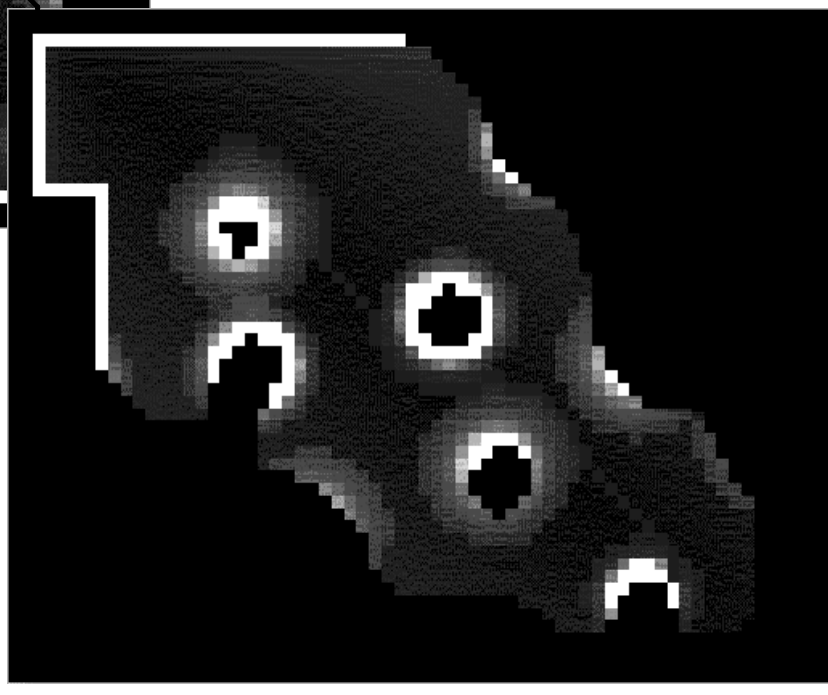

H
f(n)
g(n)
Tatsächliche Kosten

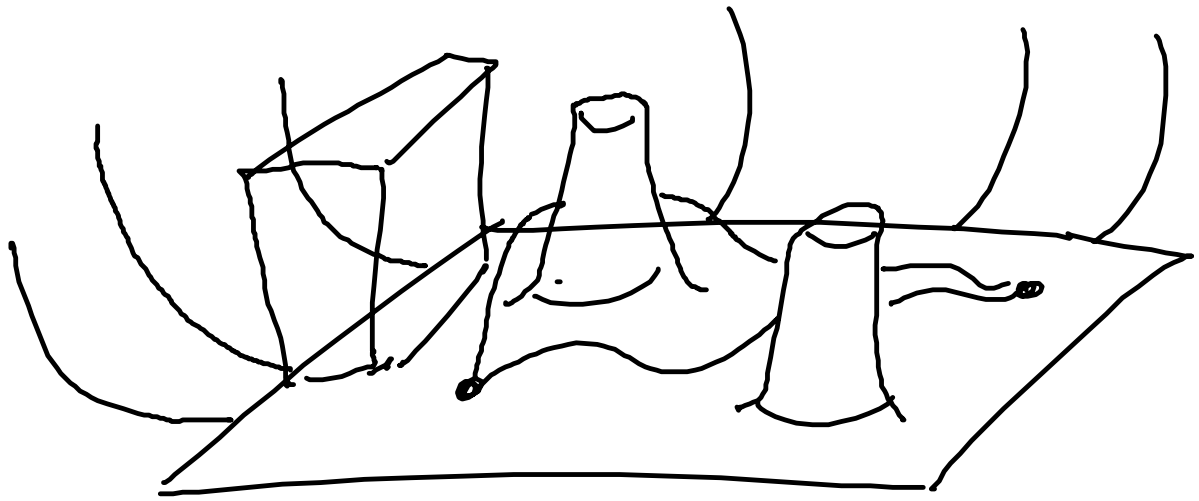

Agent:
Team 0
2





Potential Methode





Sortieren

Repräsentation

bd(2, 3, 1, 5, 6, 7, 8, 4, b)

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

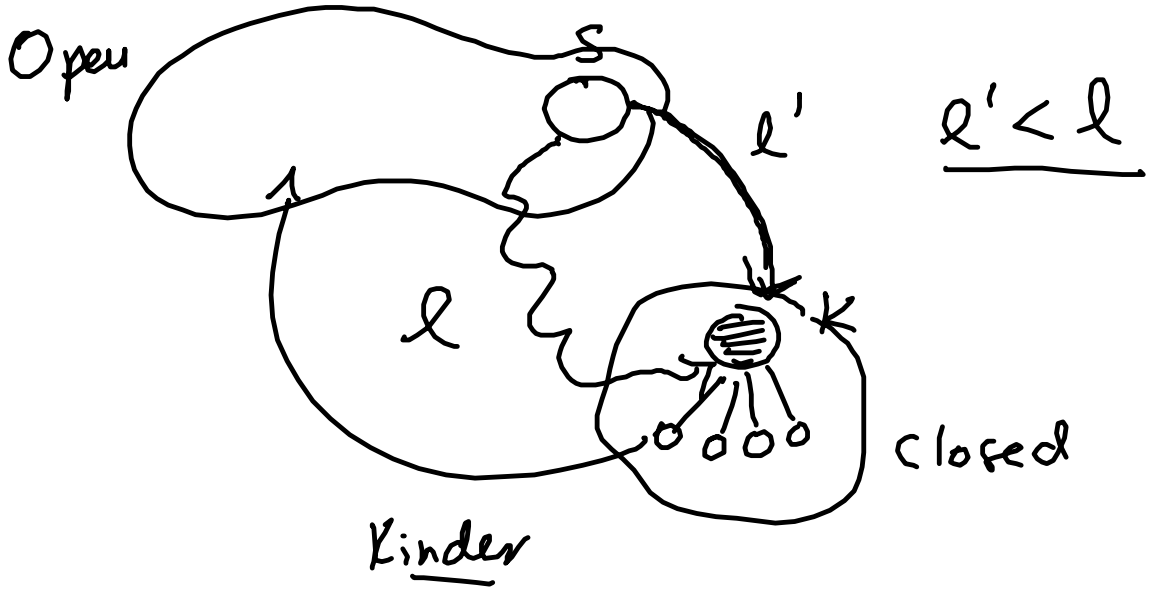
$g(n)$ ~ Tiefe

$h(n)$ Anzahl der falsch platzierte Steine

Liste von Paaren

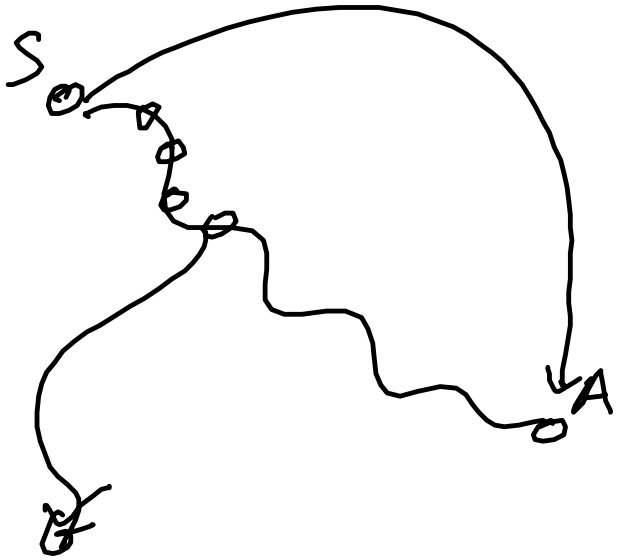
Open [[bd(....), 5], [bd(....), 6] ....]
Brett f(n)

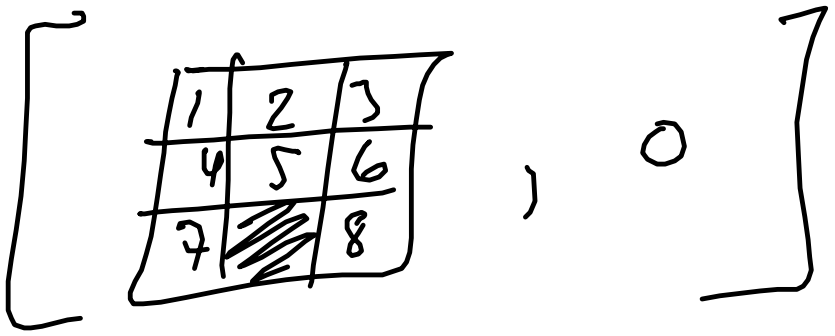
Closed [[bd(....), 7], [bd( ), 8] ....]
....




```
                Goals      Open      Closed
best(Goals, [[X,_]|_], _):-
        member(X, Goals), !, write(X).

best(Goals, [[X,T]|RestOpen], Closed):-
        member([X,T1], Closed),
        T>=T1, !,
        best(Goals, RestOpen, Closed).

best(Goals, [[X,T]|RestOpen], Closed):-
        member([X,T1], Closed),
```

```
        T < T1, !,
        kinderbest(X, T, L),
        appendm(L, RestOpen, NewOpen),
        qsort(NewOpen, Open),
        best(Goals, Open, [[X|T]|
                                Closed]).

best(Goals, [[X|T]|RestOpen], Closed):-
        write('open '), write(X),
        heuristik(X, T, FX), write(FX),
        kinderbest(X, T, L),
        appendm(L, RestOpen, NewOpen),
        qsort(NewOpen, Open),
        best(Goals, Open, [[X|T]|Closed]).
```





myqsort ( ) ← selber programmieren

```
kinderbest( X, T, L) :-
        T1 is T+1,
         setof( [Y, T1], move(X,Y), L), !.

kinderbest( -, -, [] ).
```

———— o ————

Bestensuche A^*

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

↑ Tiefe ↑ Heuristik

Echte Kosten $h(n)$

$$h^*(n) \leq h(n)$$

Heuristik die Aufwand unterschätzt

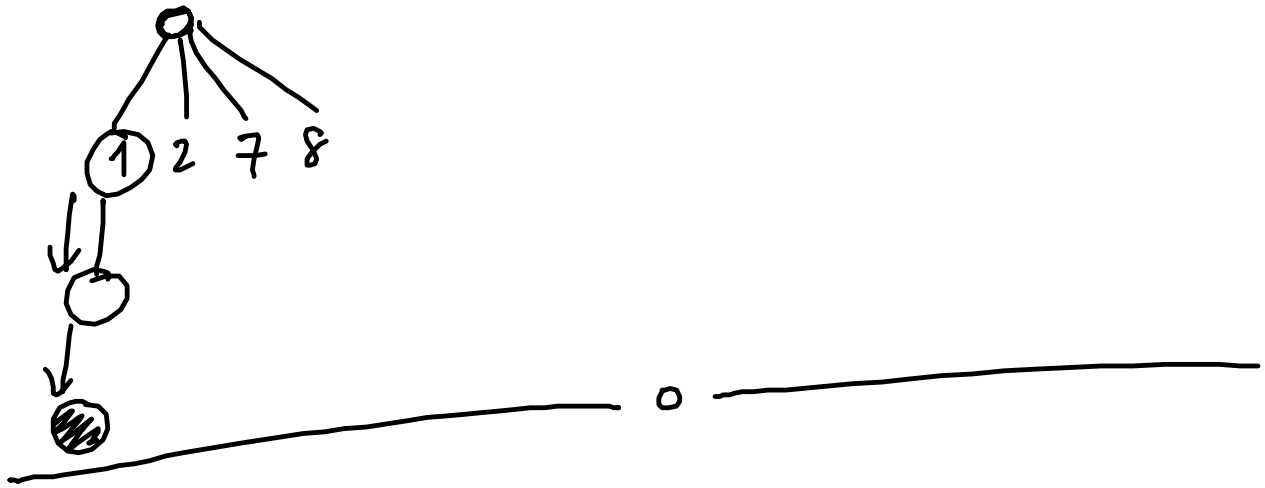
A^* ist Bestensuche mit einer Heuristik, die den echten Aufwand unterschätzt.

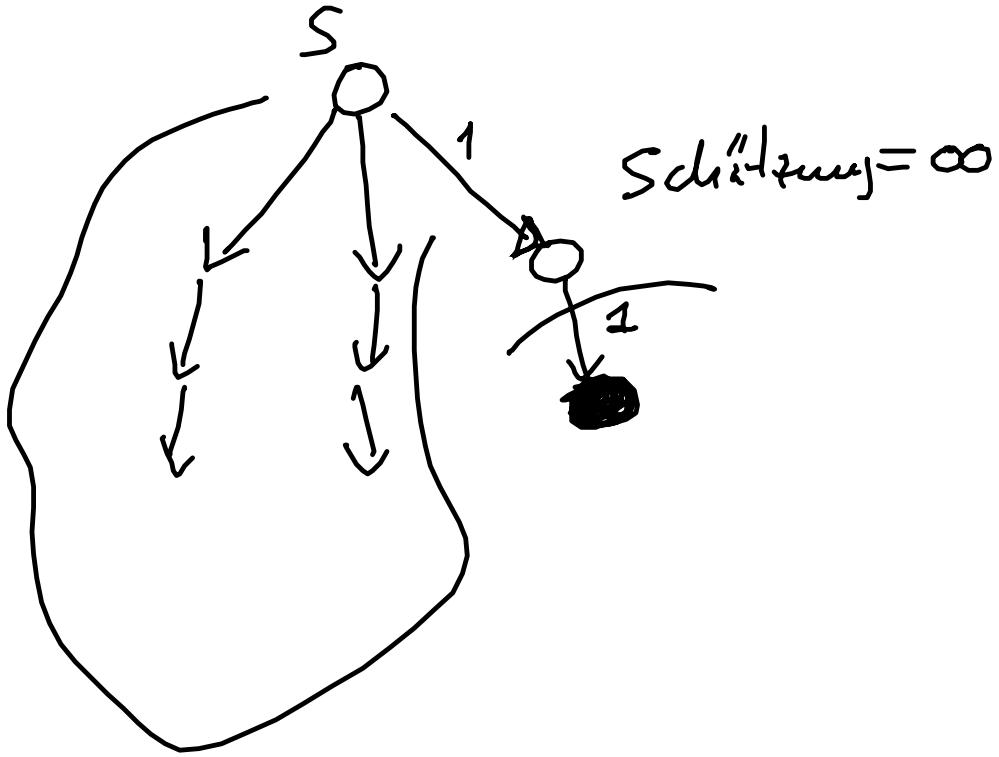
⇒ findet eine Lösung, falls sie existiert (admissible)

Fall

1 2 3

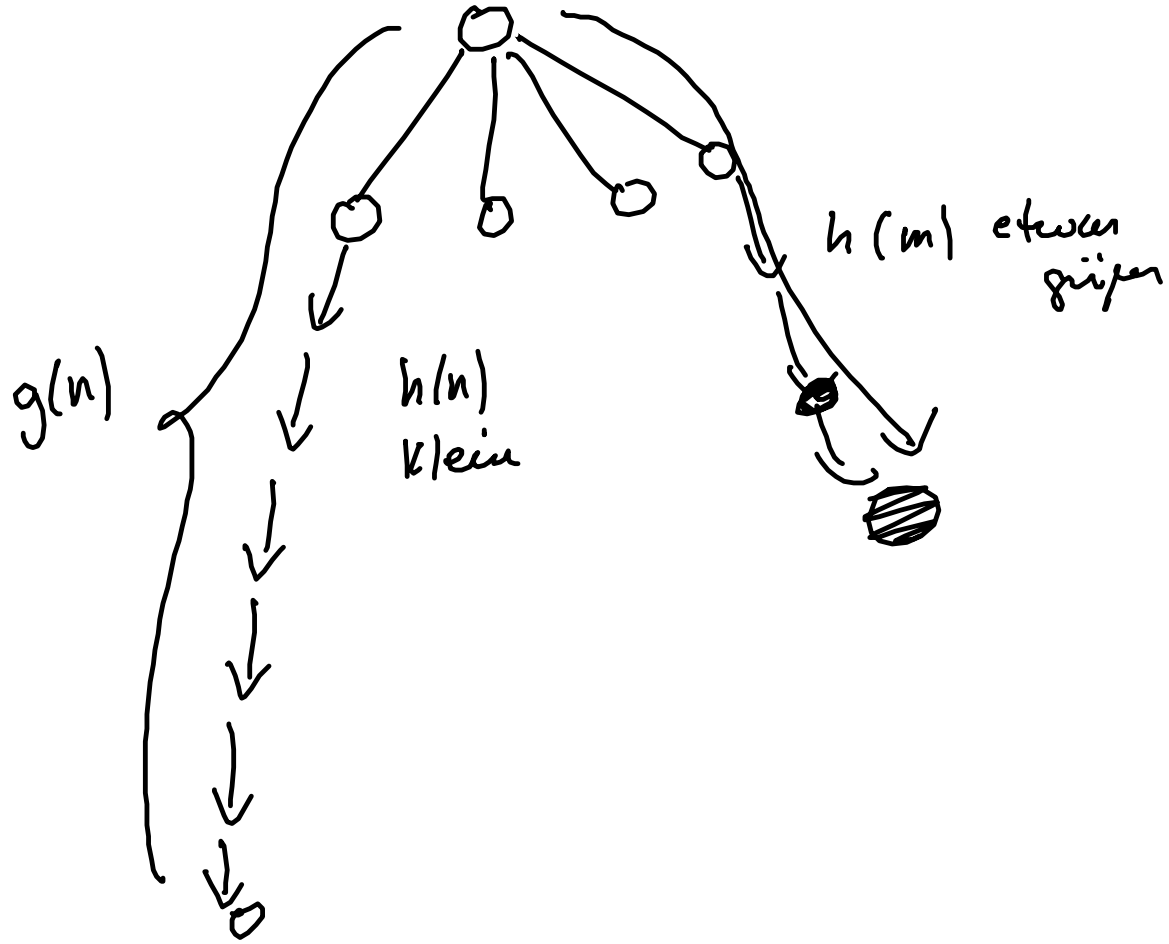
$h^*(n) = h(n)$ $h^*(n) \leq h(n)$ $h^*(n) > h(n)$



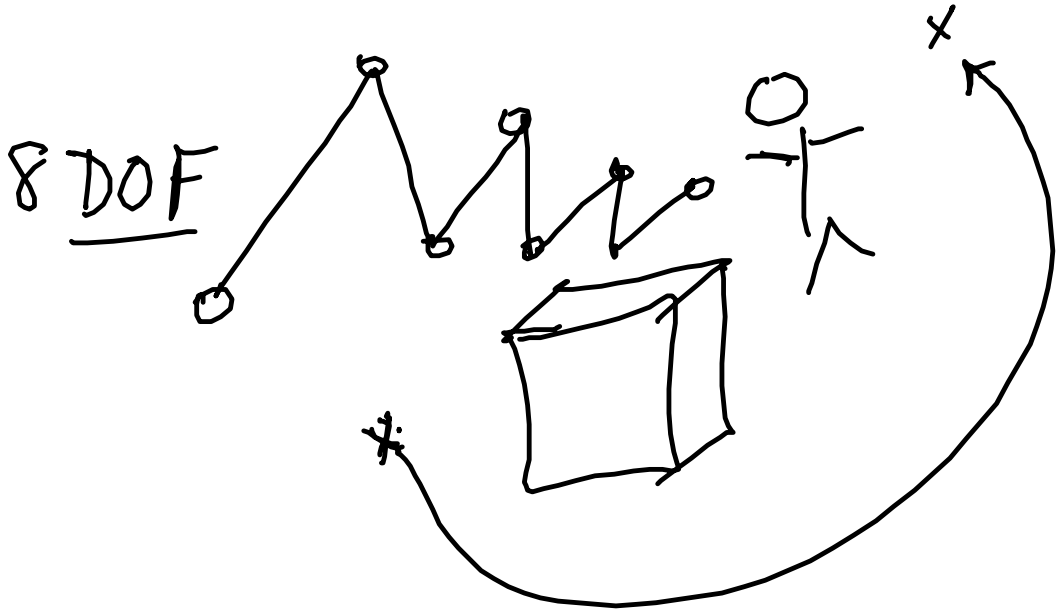


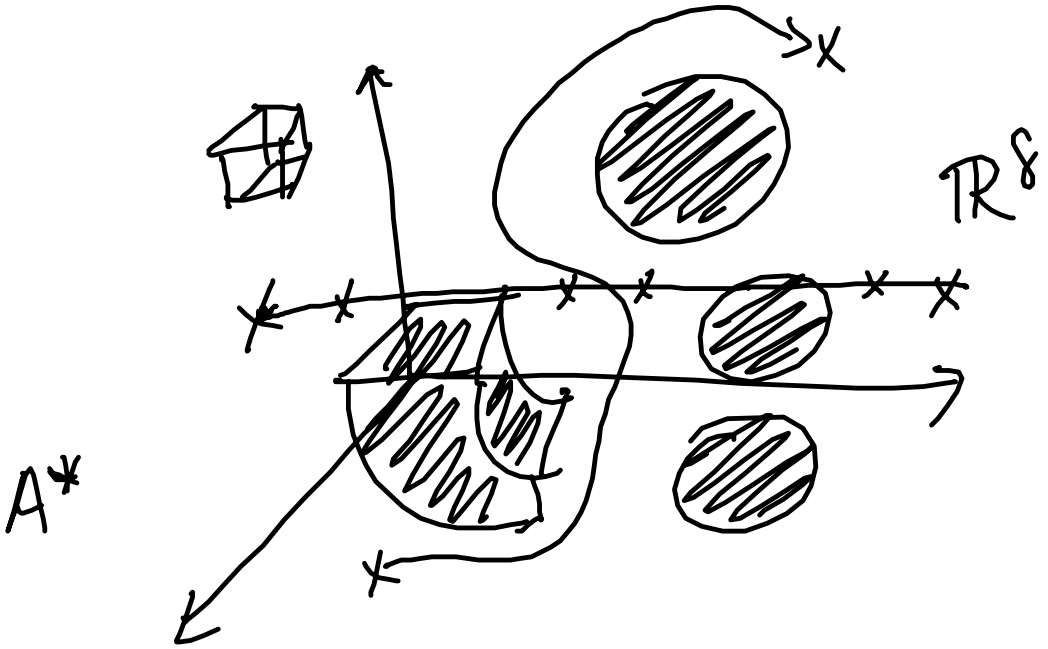
Unterschätzung

S



letzter Bsp.




$\mathbb{R}^8$
A^*