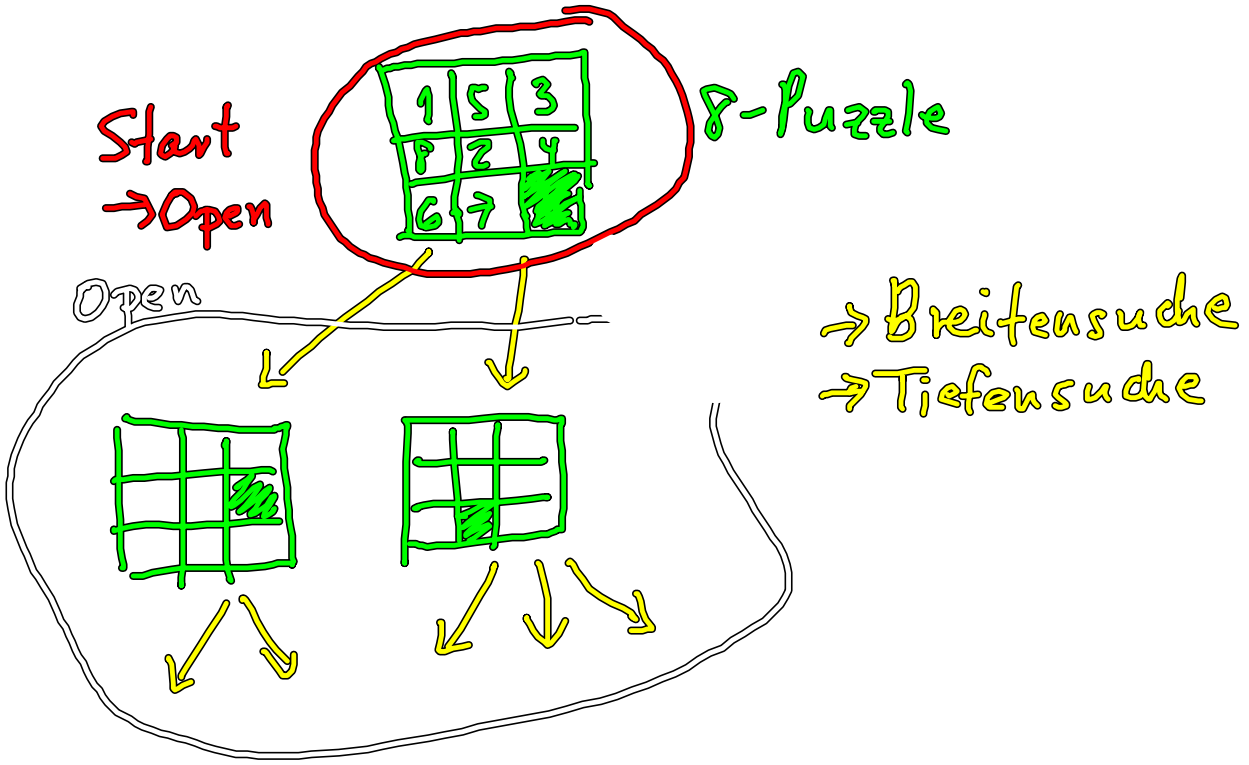
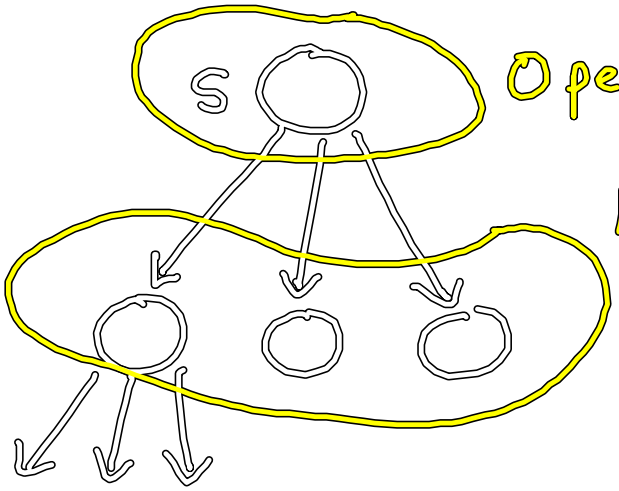
Zustandsraum



Bestensuche

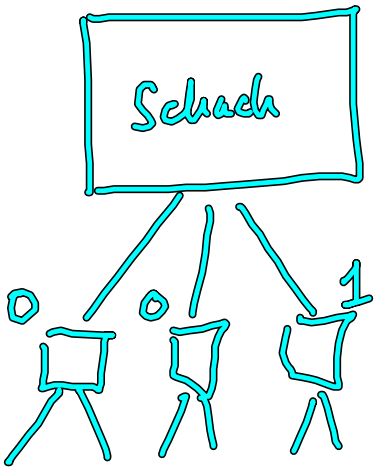


New Open := (Open − s) ∪ Kinder

sortiert

"Map
Abstandsmaß"

→ Heuristik



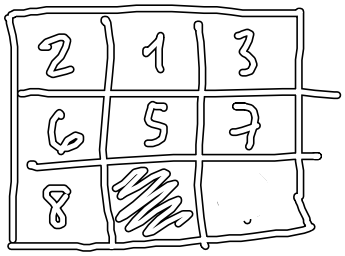
Bewertungsfunktion $\in (0, \frac{1}{2}, 1)$

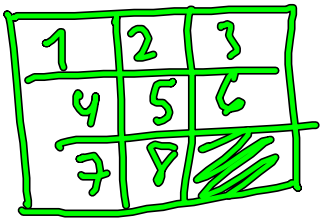
Bewertung

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

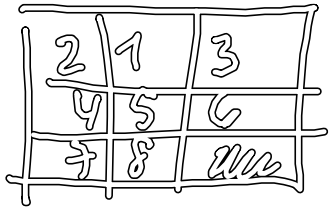
7 Fehler

Heuristik-1





2 Fehler

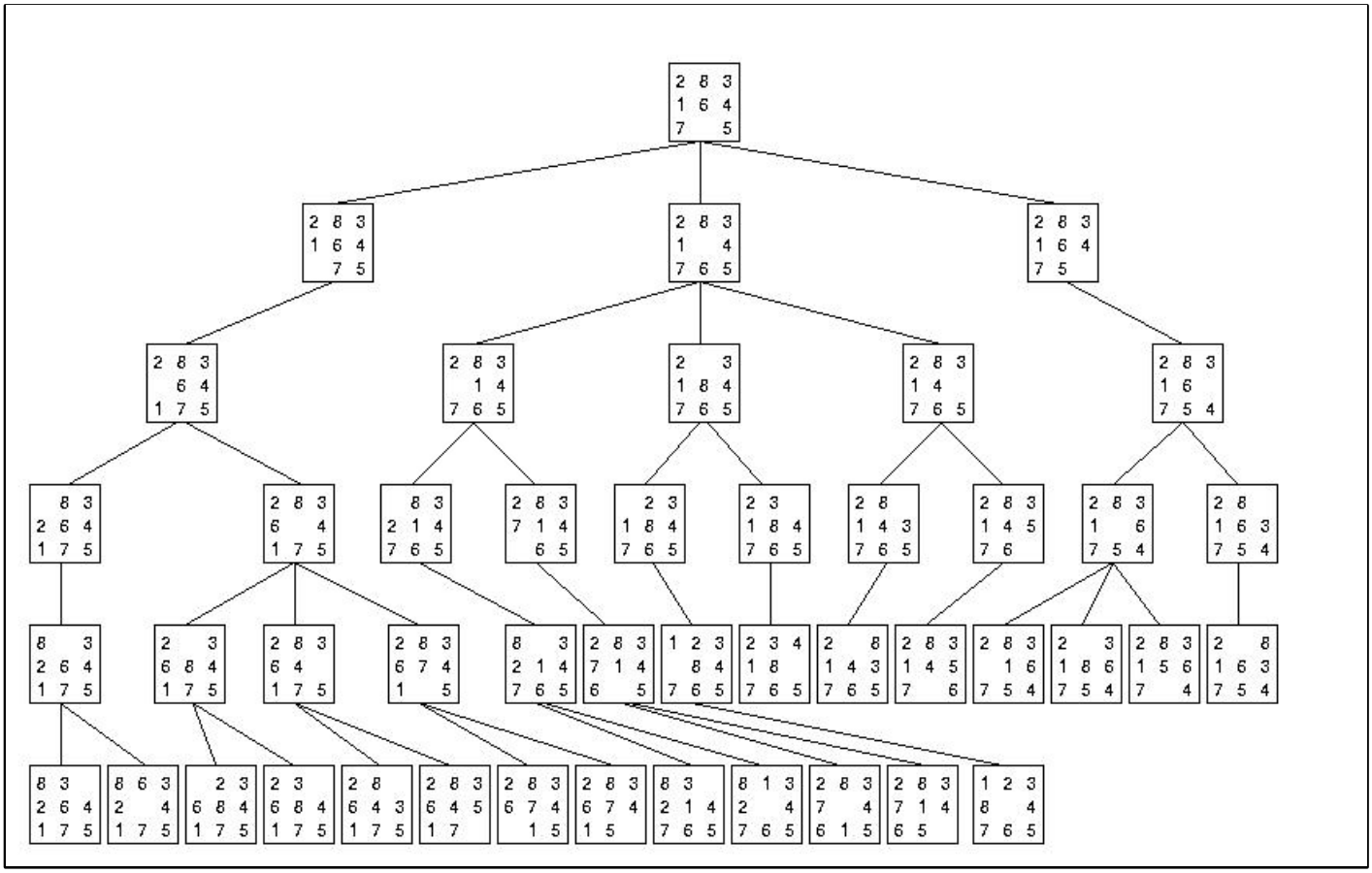


Heuristik-2

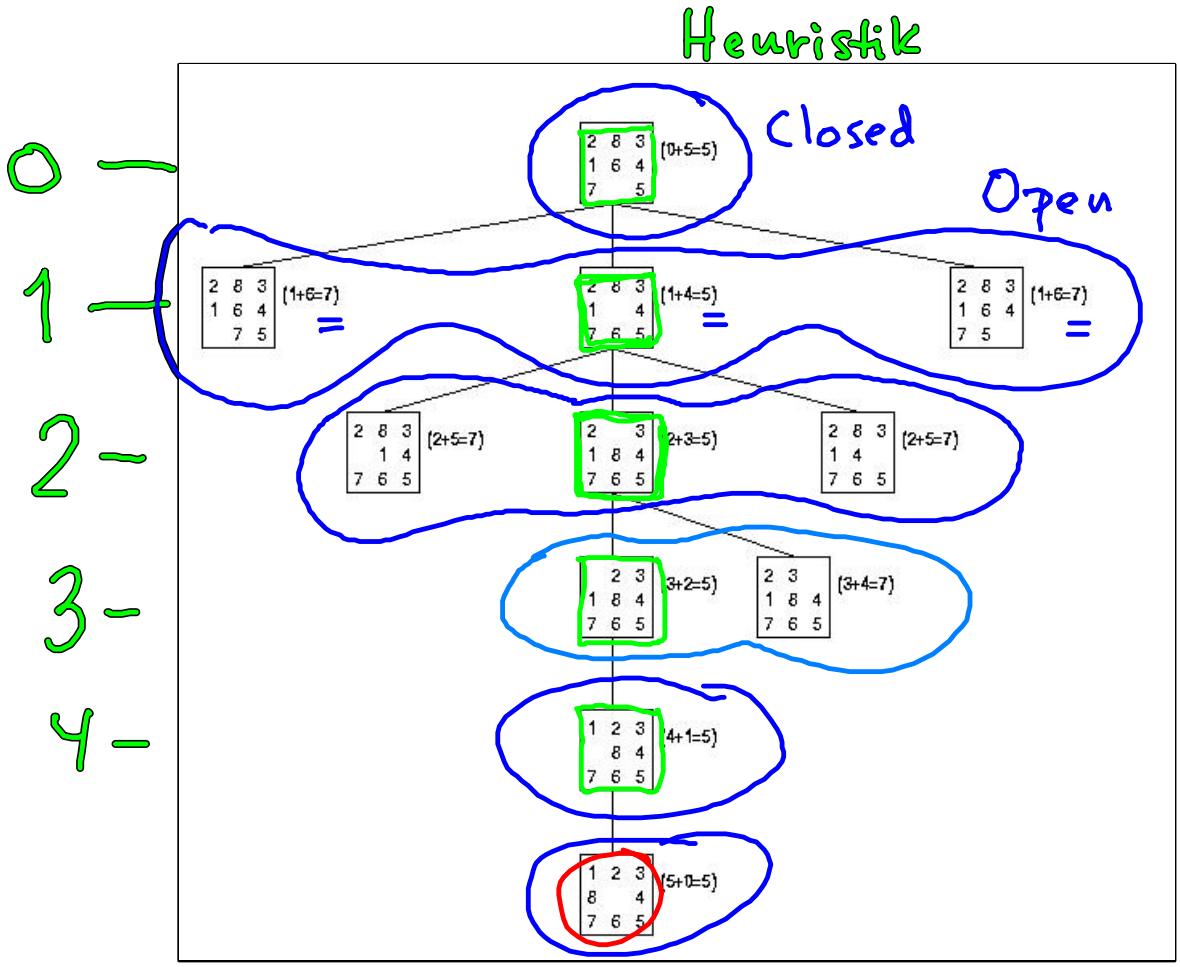
1 + 1 + 0
+ 2 + 0 + 3
+ 1 + 3 + 1

12

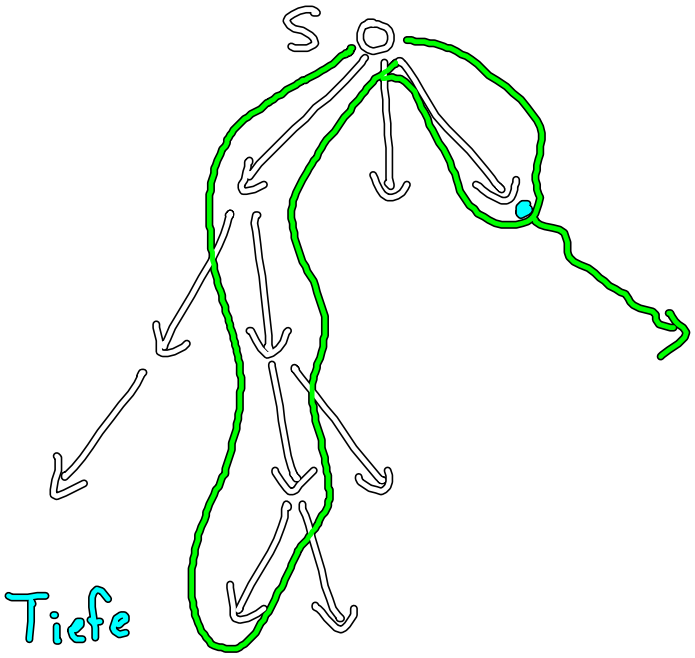
Bestensuche



Heuristik

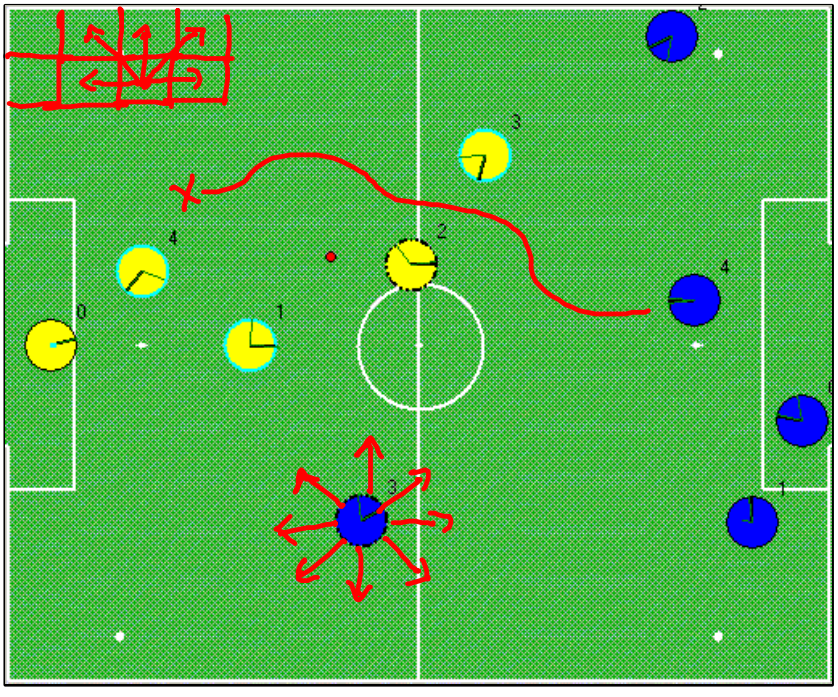


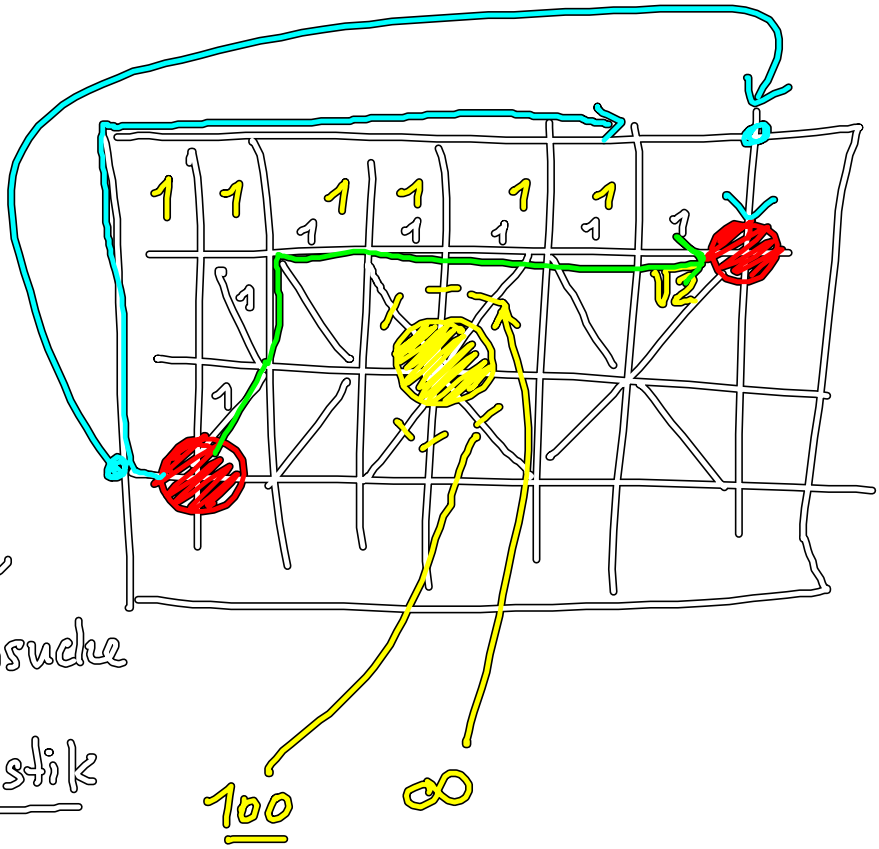
$$f(n) = g(n) + h(n)$$

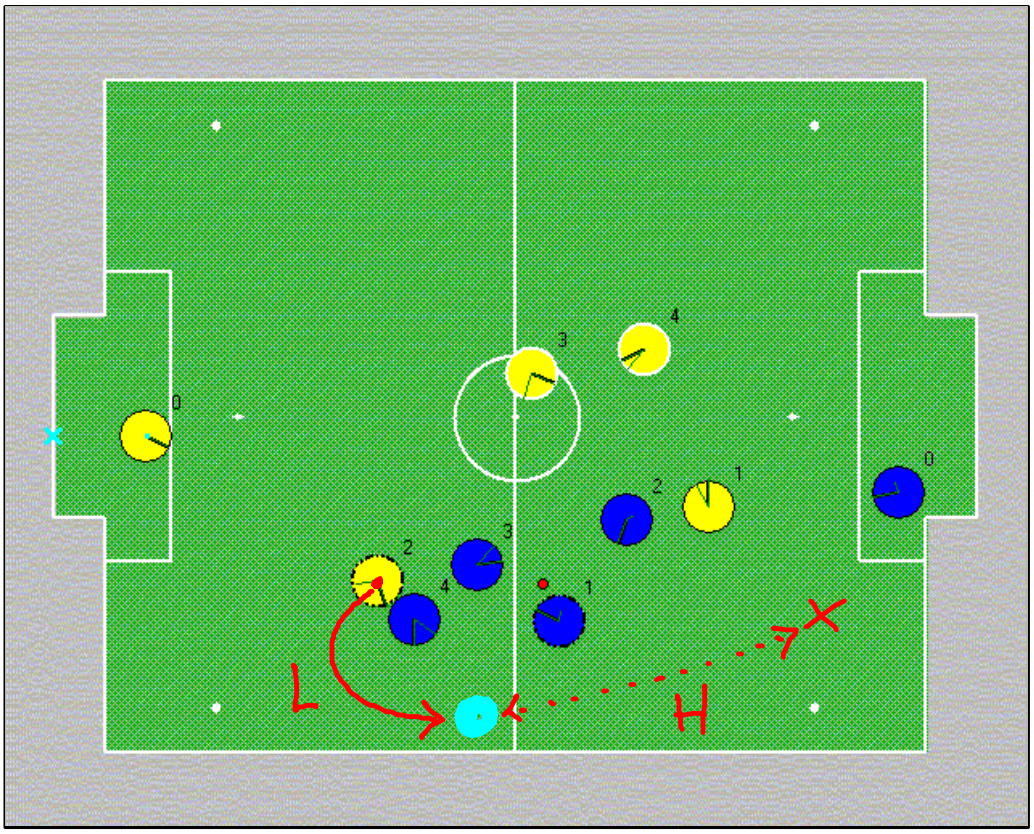


$$f(n) = g(n) + h(n)$$

zu groß

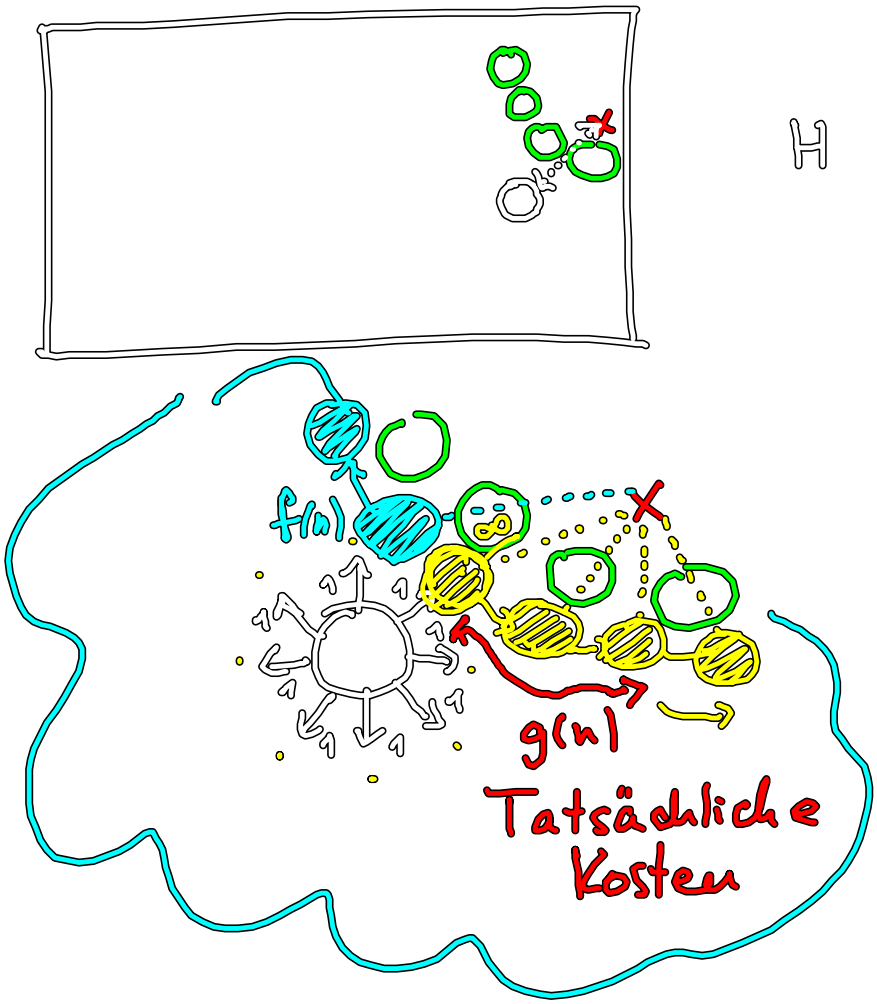


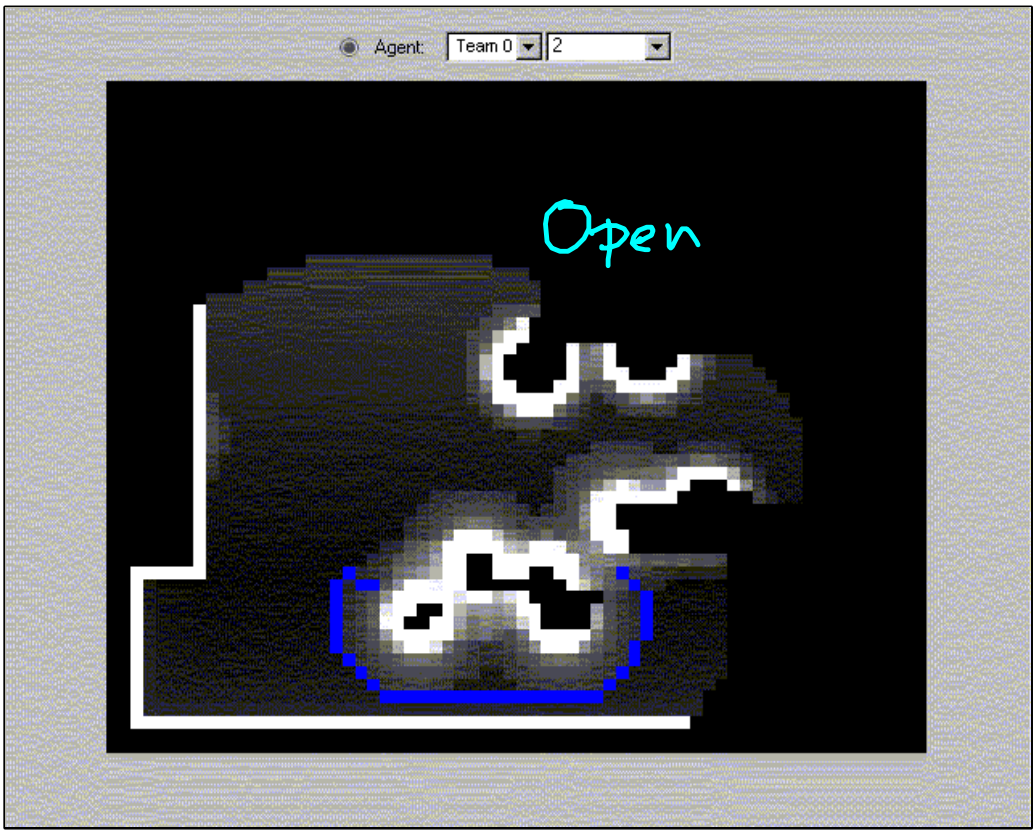


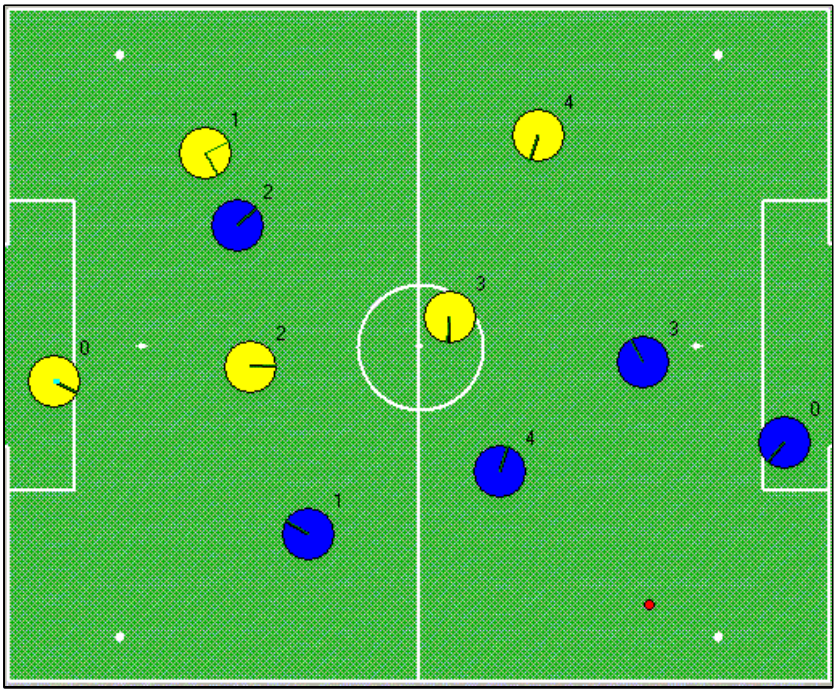


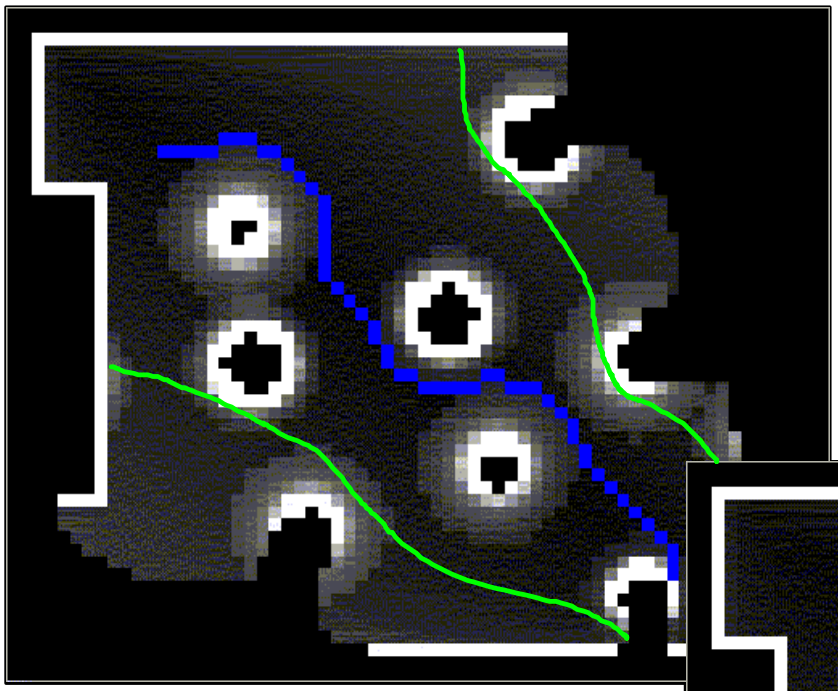
$$f(n) = g(n) + h(n)$$

L H

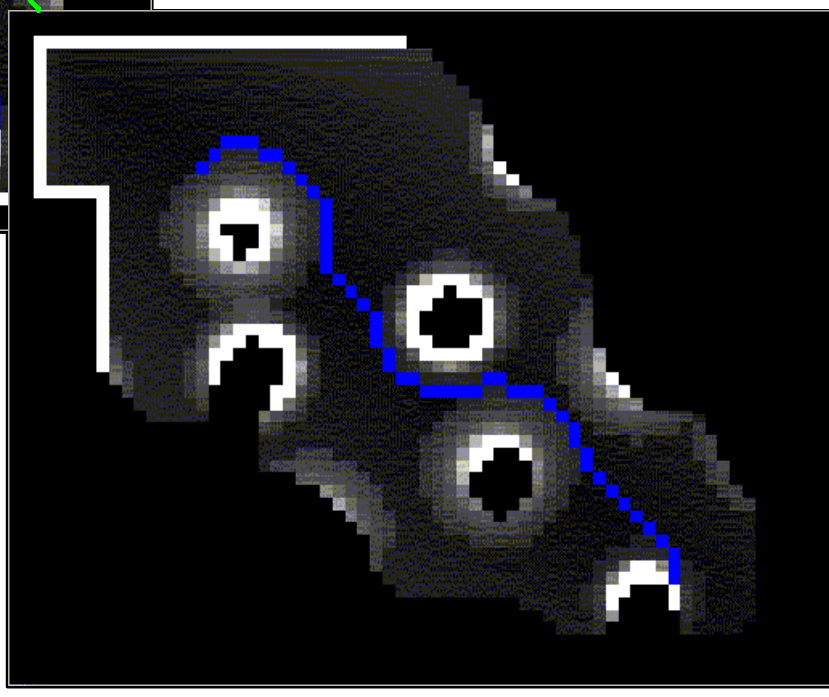

H
f(n)
g(n)
Tatsächliche Kosten

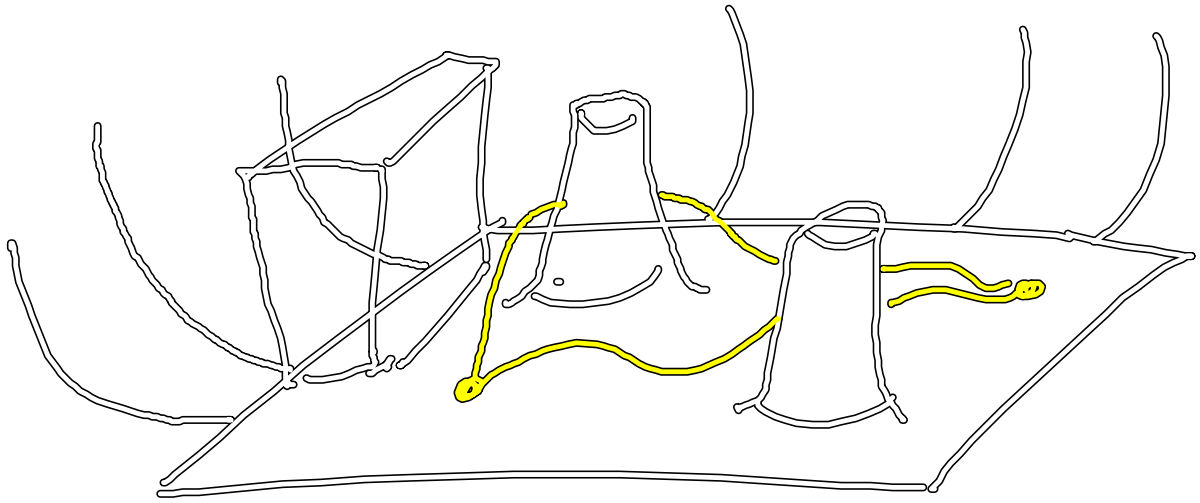

Agent:
Team 0
2
Open





Potential Methode





Sortieren

Repräsentation

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

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

$g(n) \sim$ Tiefe

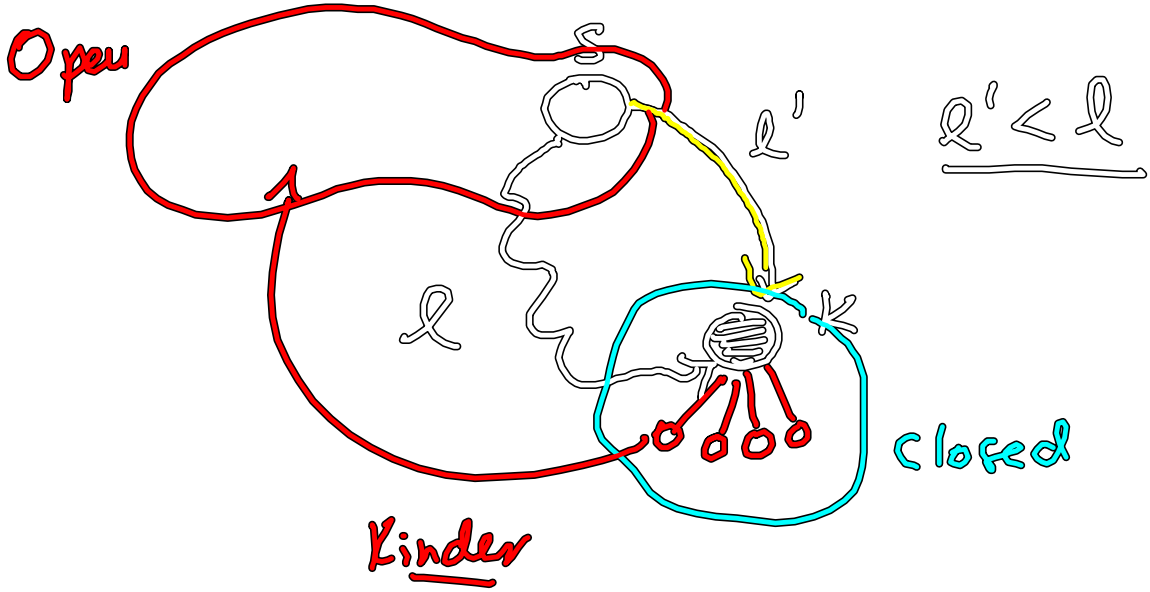
$h(n)$ Anzahl der falsch platzierte Steine

Liste von Paaren

Open [[bd(....), 5], [bd(...), 6]....]

Brett $f(n)$

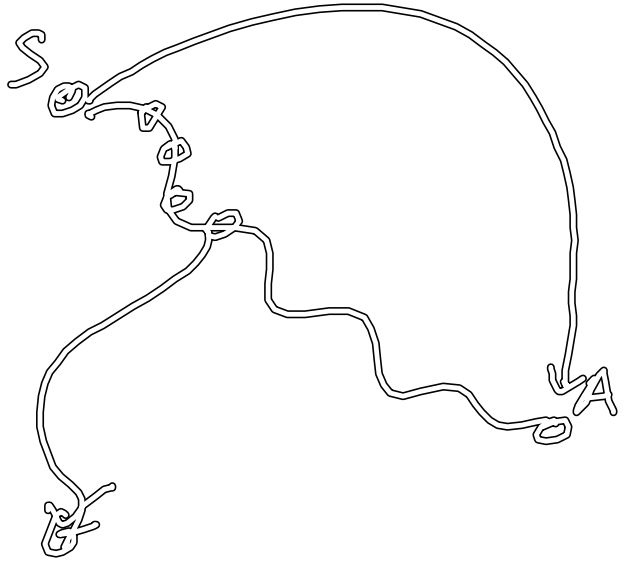
Closed [[bd(...), 7], [bd( ), 8]....]

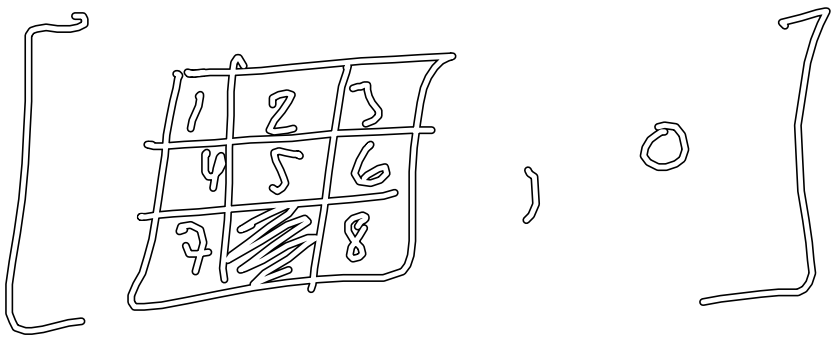


Goals Open Closed

```
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, [[_|_]|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(-, -, []).

Bestensuche A^*

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

$g(n)$: Tiefe

$h^*(n)$: Heuristik

Echte Kosten $h(n)$

$$\underbrace{h^*(n) \leq h(n)}$$

Heuristik die Aufwand unterschätzt

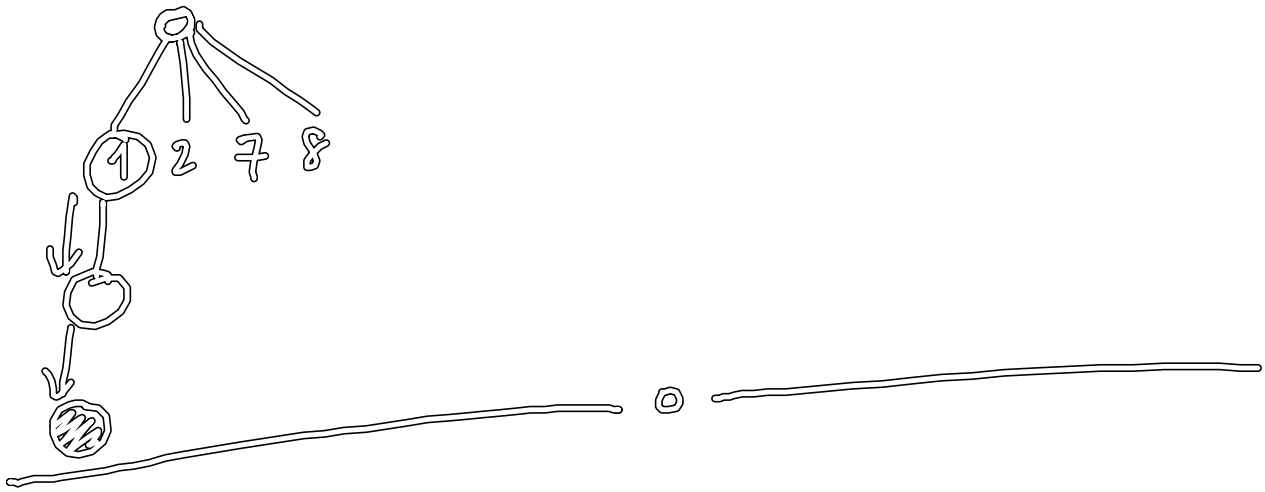
A^* ist Bestensuche mit einer Heuristik, die den echten Aufwand unterschätzt.

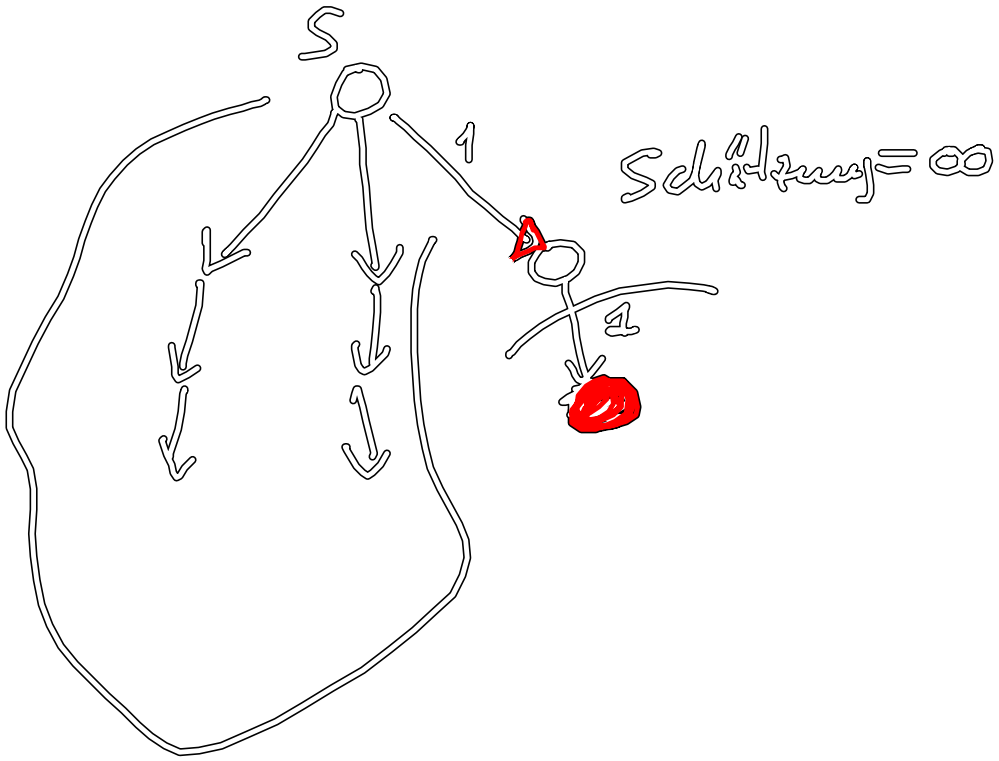
$\Rightarrow$ findet eine Lösung, falls sie existiert (admissible)

Fall

1 2 3

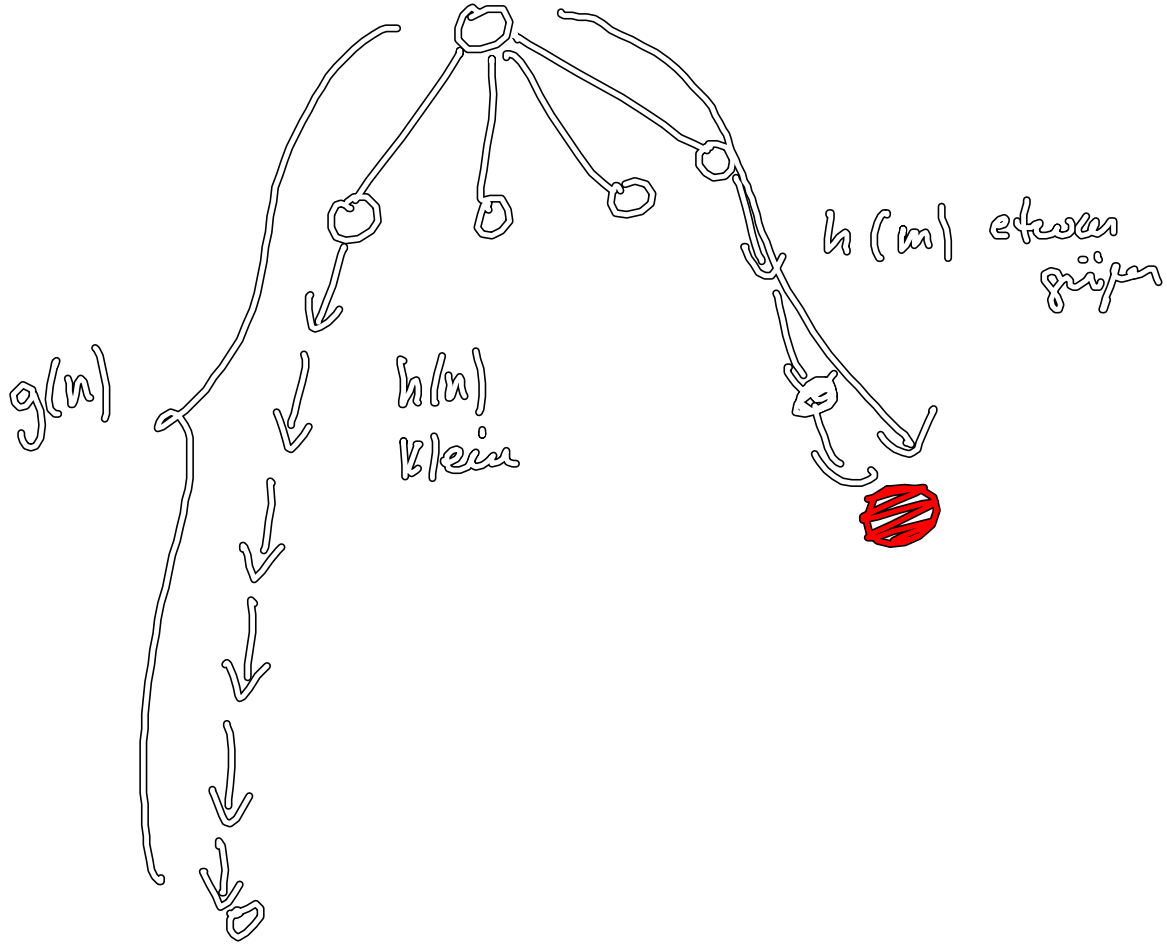
$h^*(n) = h(n)$ $h^*(n) < h(n)$ $h^*(n) > h(n)$



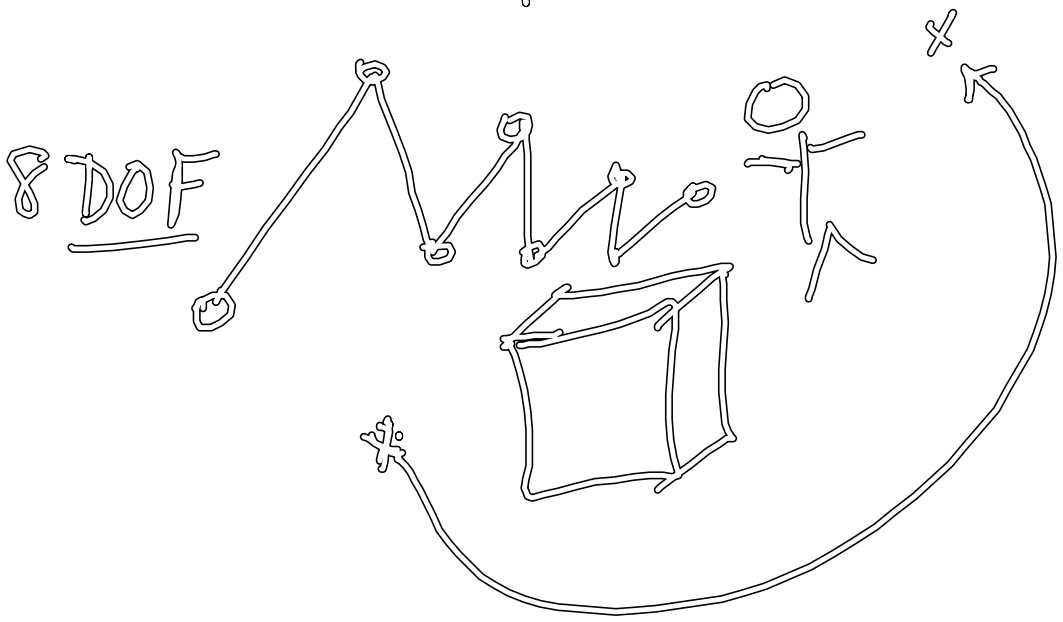


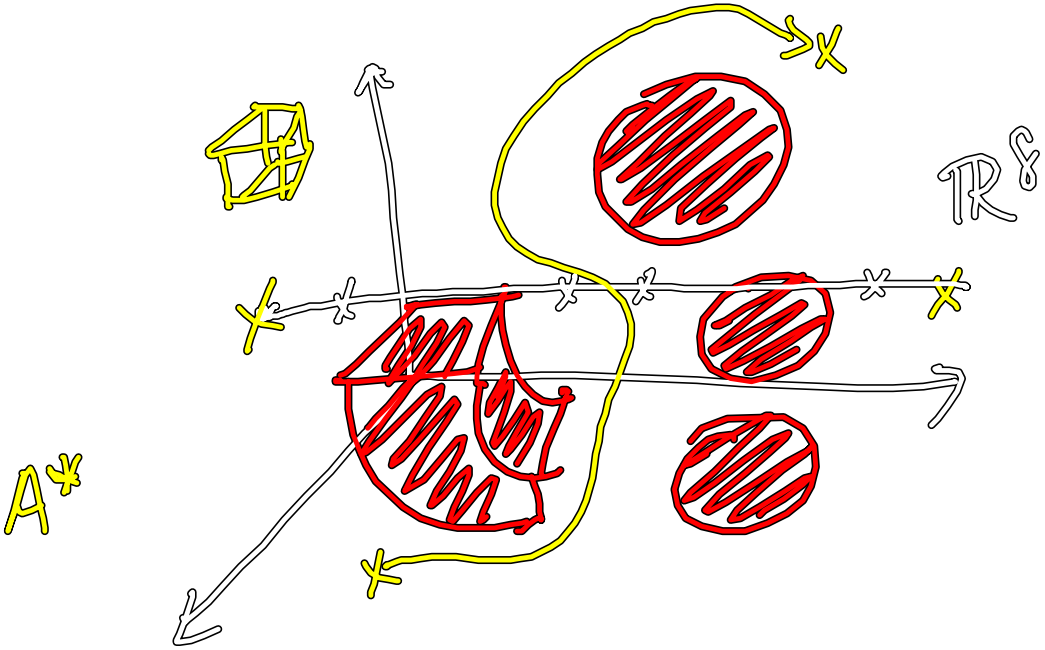
Unterschätzung

S



letzter Bsp.




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