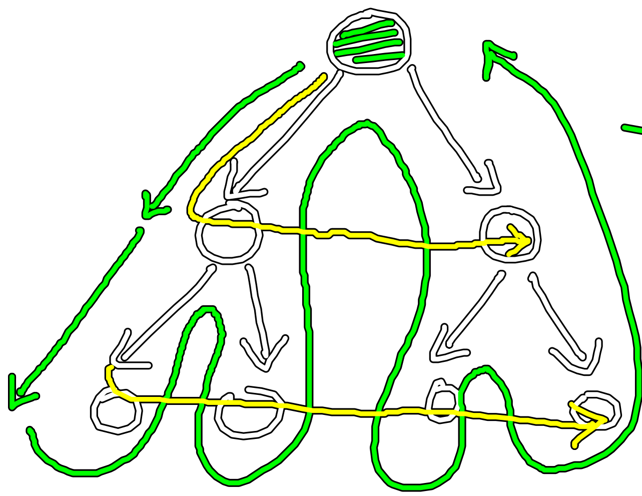
A* - Algorithmus



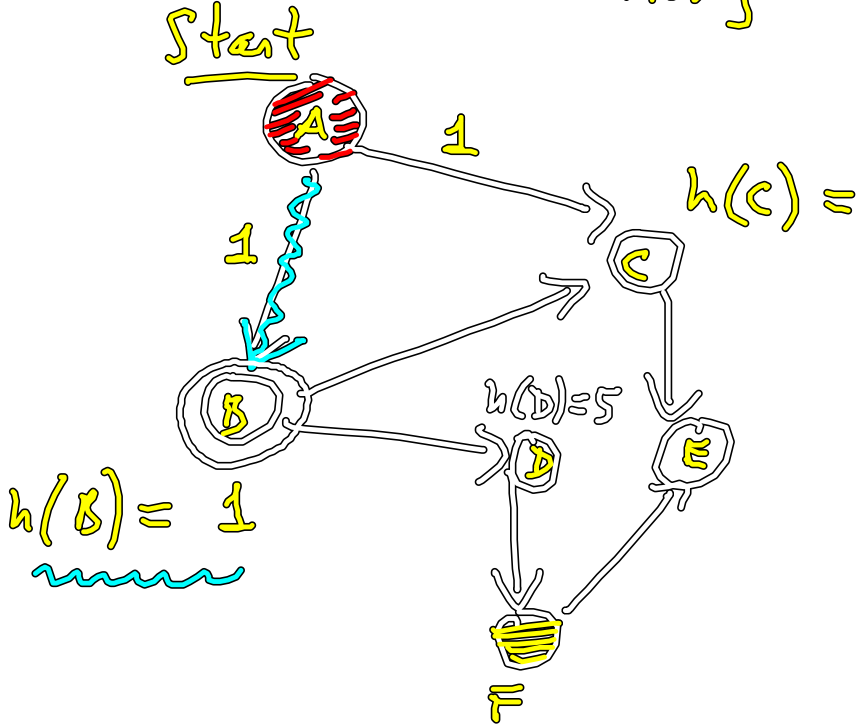
1) Sätze - Beweisen
   → Prolog (Tiefensuche)

2) Puzzles

3) Pfadplanung

A*   Variante "Best first search"
   Heuristik (optimistisch)

Bewertung

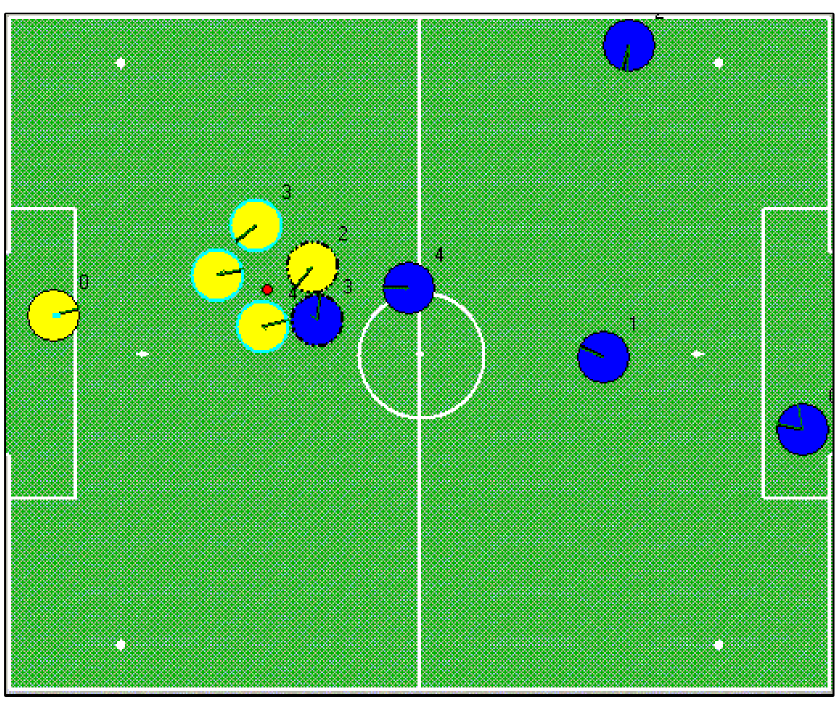


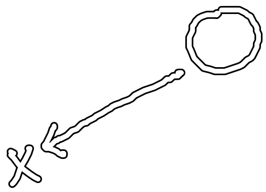
Open = { A }

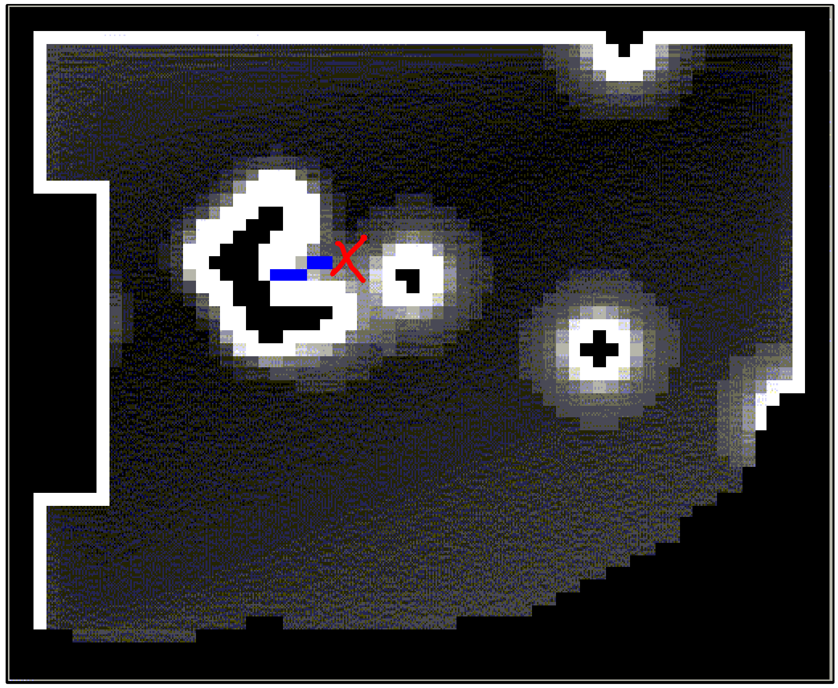
Open = { (B, 1), (C, 2) }

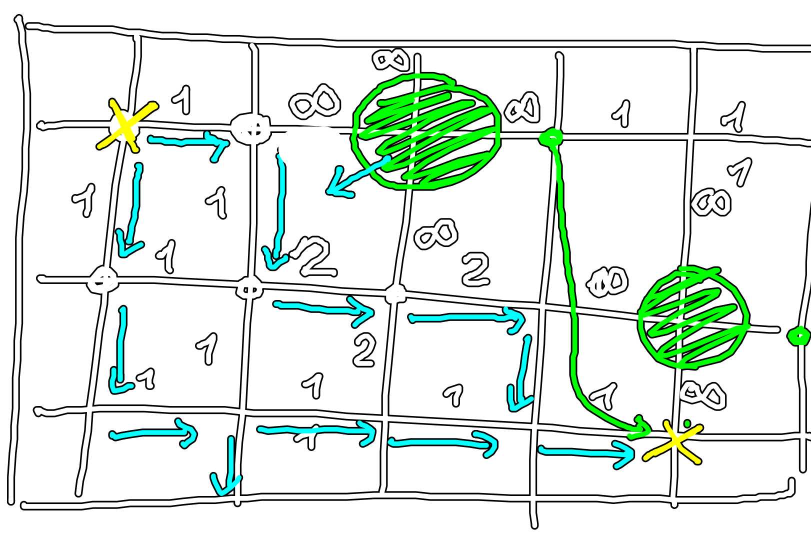
Open = { (C, 2) }

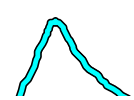
Open = { (C, 2), (C, 2), (D, 5) }

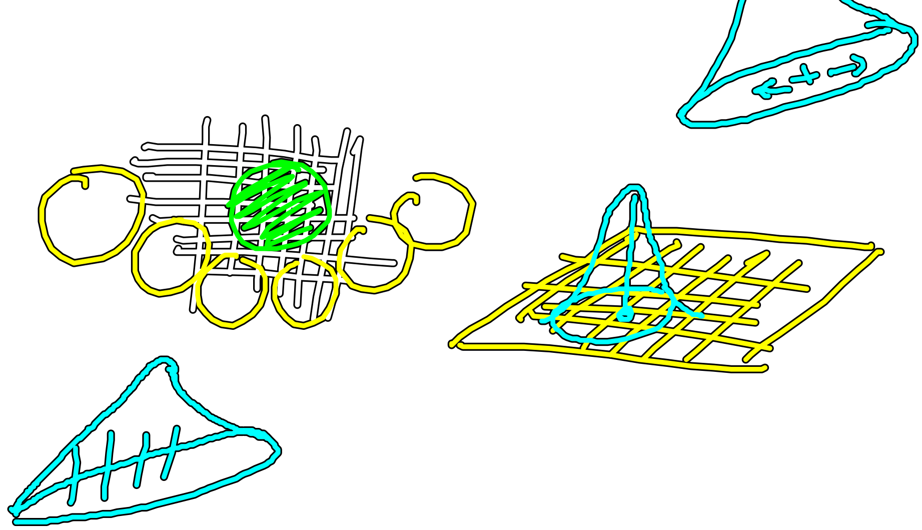


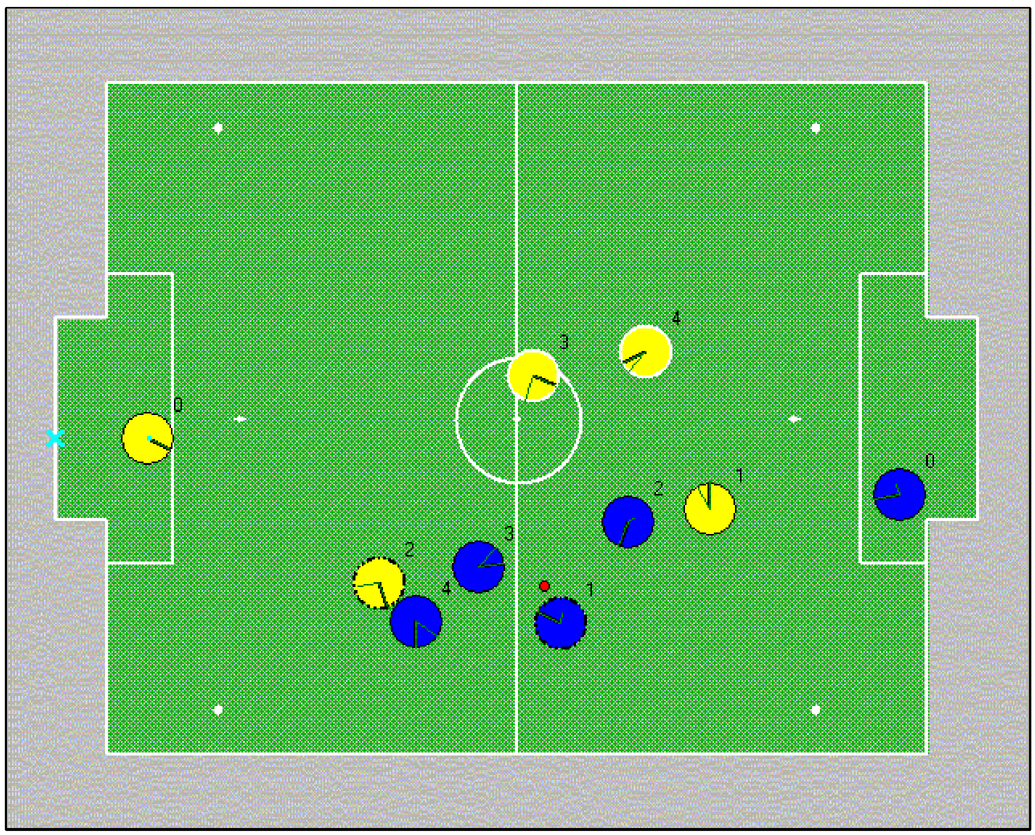


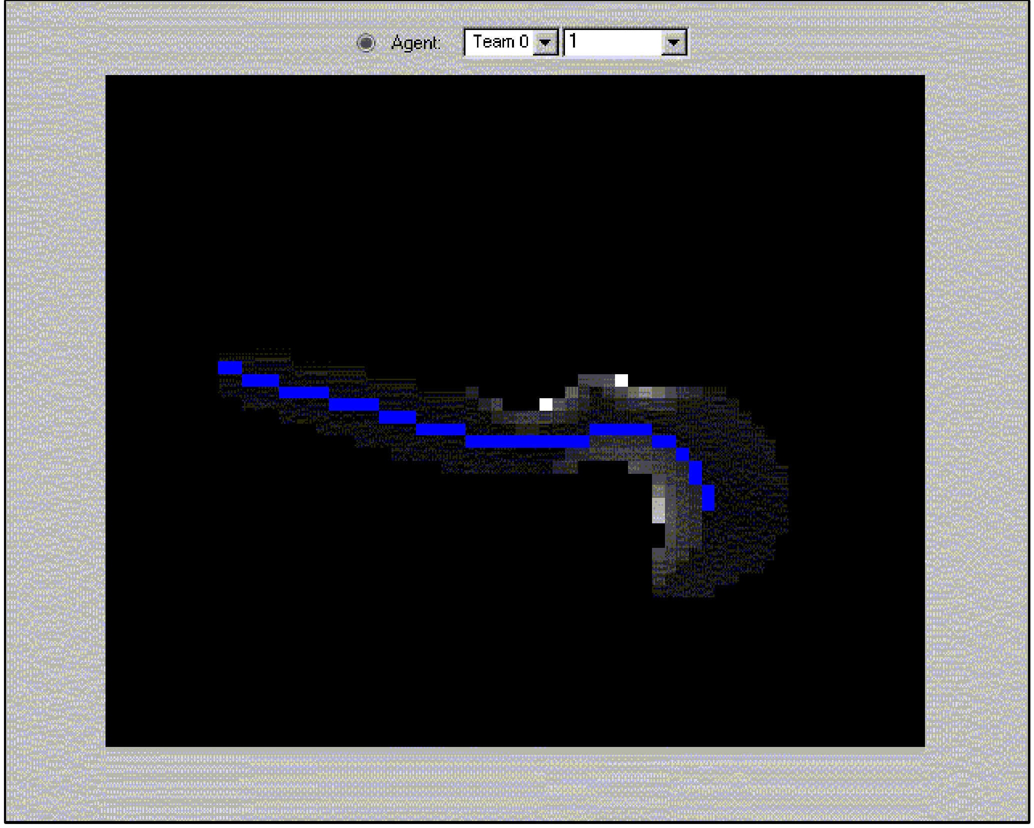


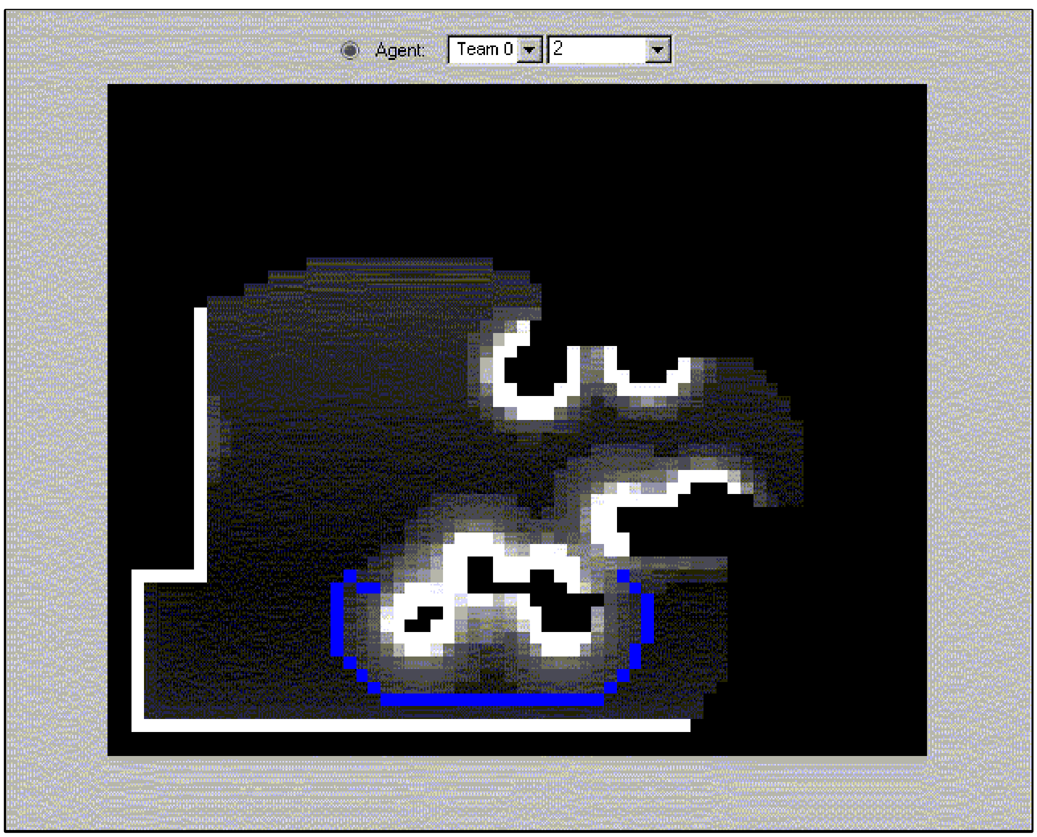


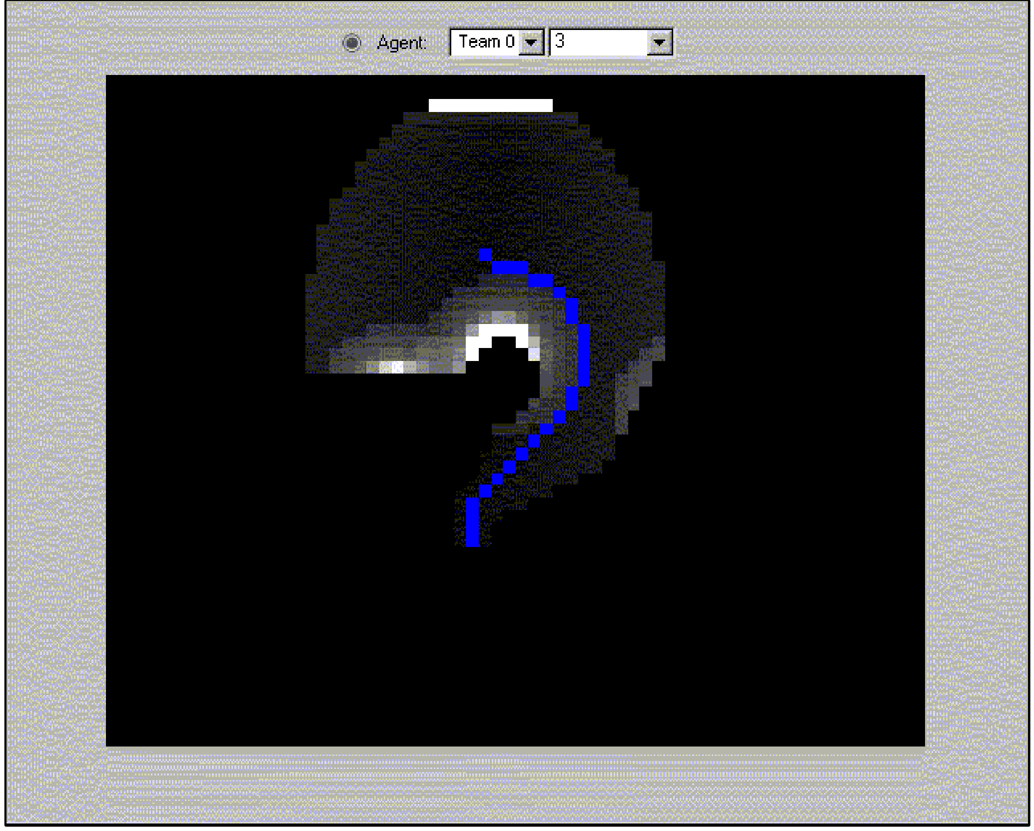




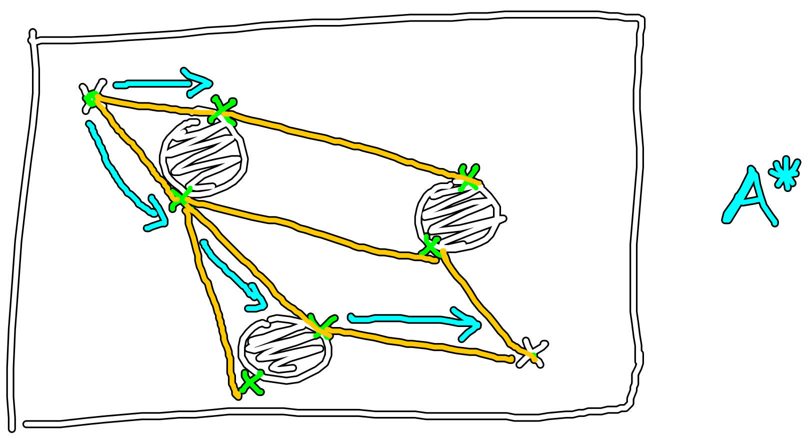



Agent:
Team 0
1


Agent:
Team 0
2



Visibility Graphs



$A^* = \text{Bestensuche}$

$$\text{Kostenfunktion}(n) = \overbrace{\text{echte Kosten}(n)}^{\text{vom Start}} + \underbrace{\text{Heuristik}(n)}_{\text{bis zum Ziel (Abschätzung)}}$$

↑ Knoten

Heuristik(n) $h(n)$

$$\underbrace{h(n)}_{\text{optimistisch}} \leq \text{echte Kosten bis zum Ziel}$$

"<u>acceptable</u>"

<u>A^*</u>

Open = {}   (Prioritätschlange)

Start $\rightarrow$ Open $\{(\text{Start}, \text{Kosten}(\text{Start})\}$

$\text{Kosten}(\text{Start}) = 0 + h(\text{Start})$

1) Open == {} ?

$\rightarrow$ Keine Lösung

2) Beste Knoten aus Open nehmen

Nennen wir ihn $\underline{\underline{n}}$ Kosten(n)

3) n == Zielknoten ?

$\rightarrow$ fertig

4) Kinder in die Prioritätschlange

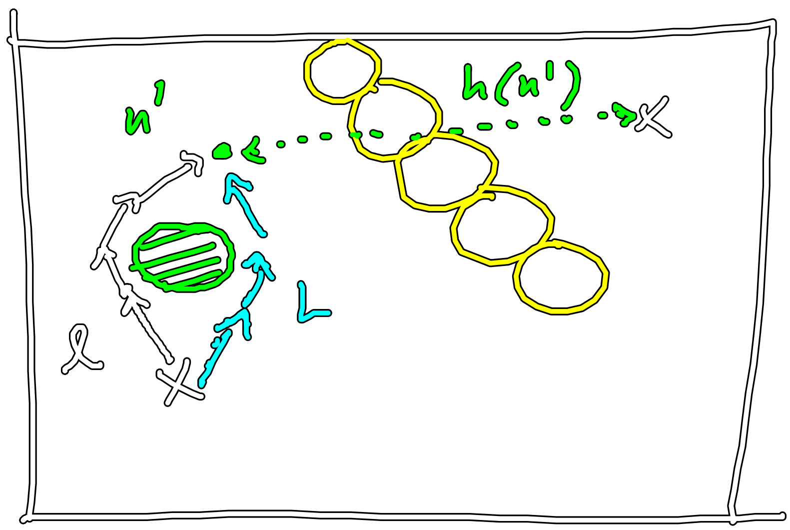
Für jedes Kind:

- n' ist ein Kind von n
- $\text{Kosten}(n') = \text{echte\_Kosten}(n') + h(n')$
- noch nicht gesehenes Kind: $(n', \text{Kosten}(n')) \rightarrow \text{Open}$

$n' \notin$ Open

schon vorher gesehen $n' \in$ Open $\left\{ \begin{array}{l} (n', \text{Kosten}'(n')) \text{ kennen wir} \\ \text{Kosten}(n') < \text{Kosten}'(n') \\ \quad \text{ja: ersetze } (n', \text{Kosten}(n')) \\ \quad\quad \text{in Open} \\ \quad \text{sonst: Open unverändert} \end{array} \right.$

GOTO 1

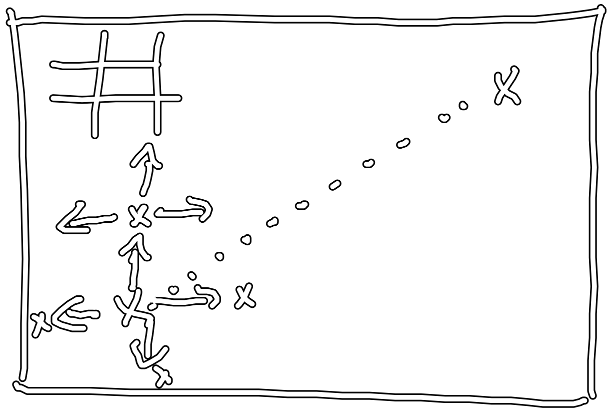


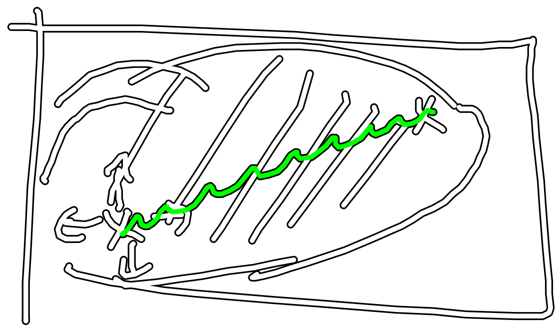
$l + h(n')$

$L + h(n')$

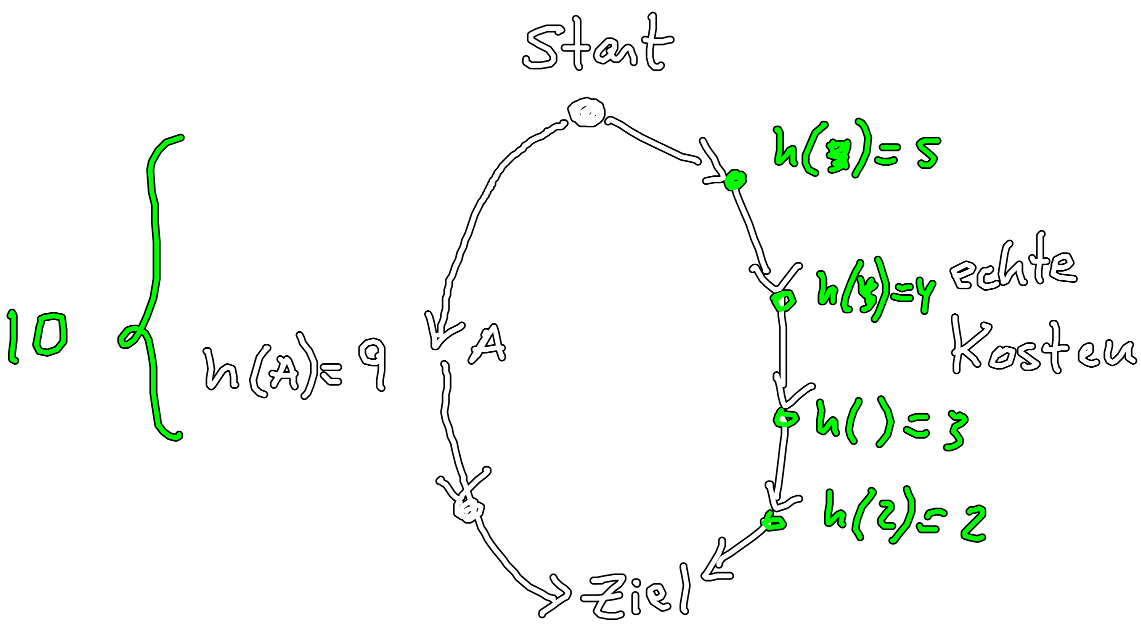
A*

$h(n') = 0$

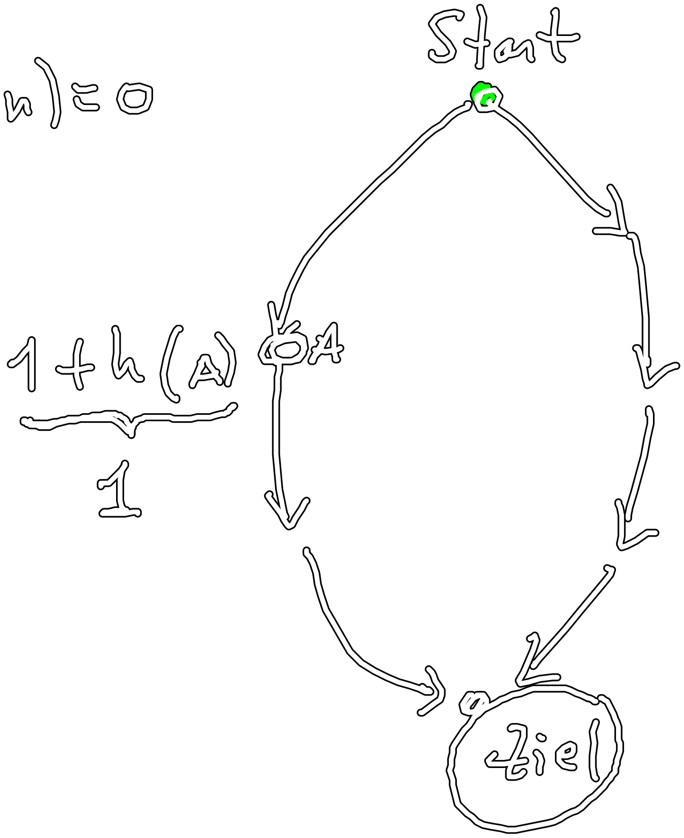




Warum optimistisch?

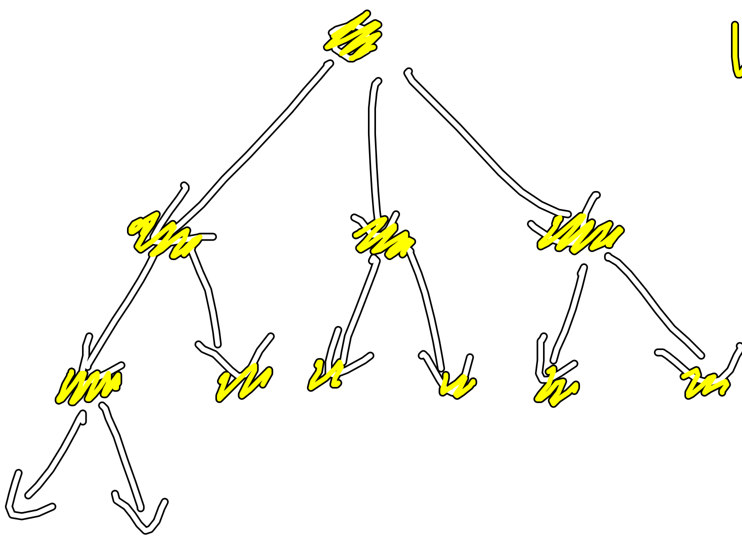


$h(n)=0$



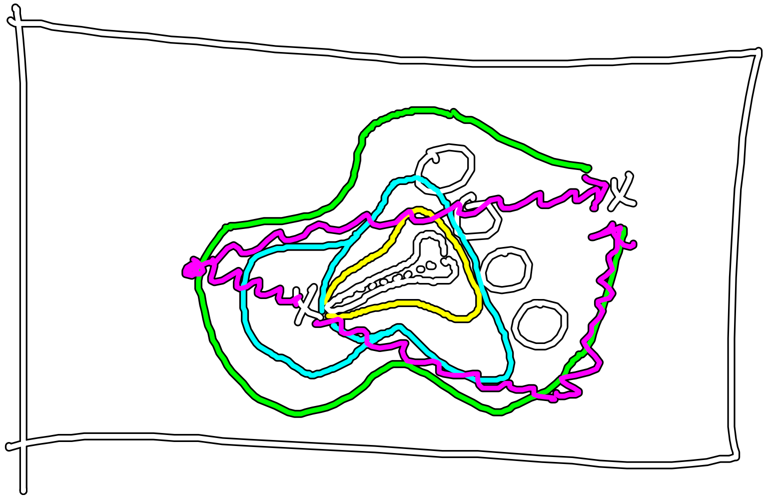
Kosten = K

$\{(Ziel, K+0)\}$

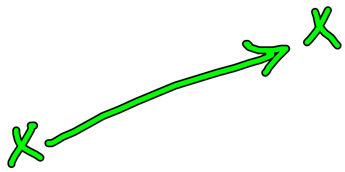


$h(n)=0$

Breitensuche

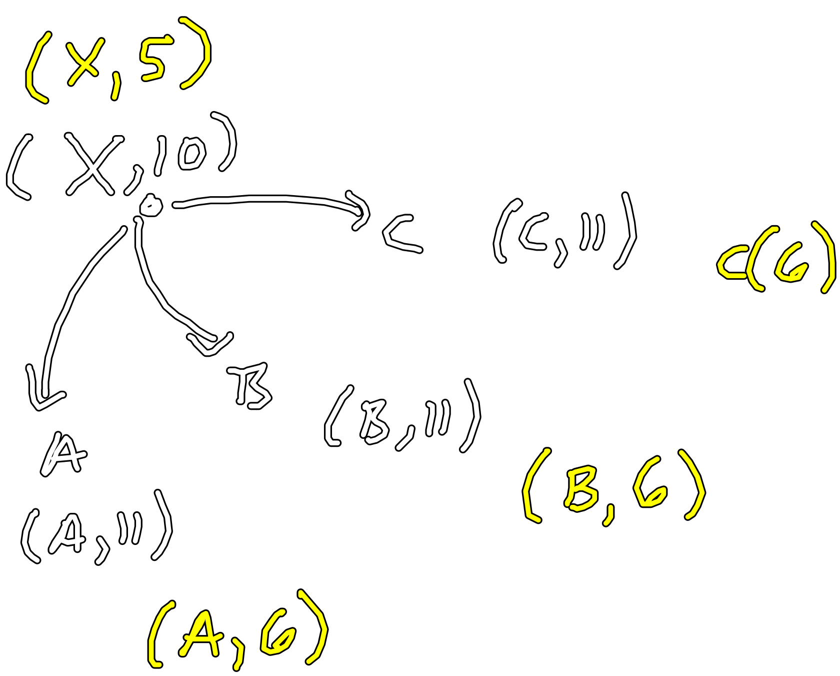


A*



```
besta(Goals, Open, Closed)

besta(goals, [[X,_]|_], _) :-
    member(X, goals), !,
     write('sol='), write(X).

besta(goals, [[X,T]|RestO], Closed) :-
   member([X,T1], closed),
```

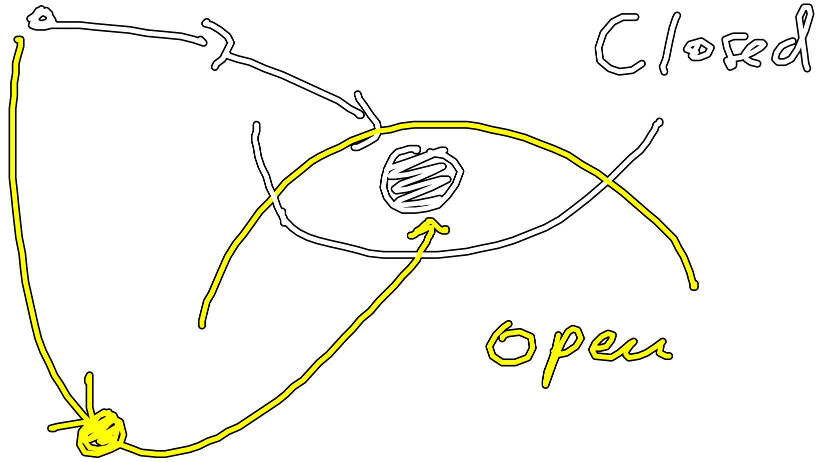
```
    T >= T1, !,
    besta(Goals, RestO, Closed).

besta(Goals, [[X,T]|RestO], Closed):-
    member([X,T1], Closed),
    T < T1, !,
    kinderbest(X, T, L),
    appendm(L, RestO, NewOpen),
    qsort(NewOpen, Open),
    besta(Goals, Open, Closed).
```



Kinderbest(X, 5, L)

[[C,6], [B,6], [A,6]]

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



Goal

1	2	3
4	5	6
7	8	b

Heuristik

Move-Generator

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

⋮

Heuristik

$$h\left(\begin{array}{|c|c|c|}\hline 2 & 3 & 1 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & b \\ \hline\end{array}\right) = 3$$

Manhattan-Abstand

$$h\left(\begin{array}{|c|c|c|}\hline 2 & 3 & 1 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & b \\ \hline \end{array}\right) = 4$$

$$h\begin{pmatrix} b & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{pmatrix} = 4 + 4$$

$$= 8$$