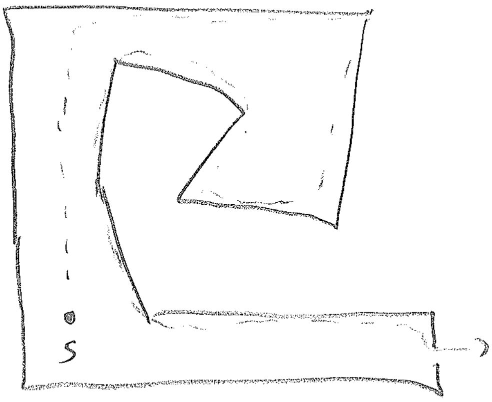
Back to a classic: How to escape from a maze?
→ Pledge algorithm (John Pledge, 12 years old
Abelson/diSessa '80)

the model:

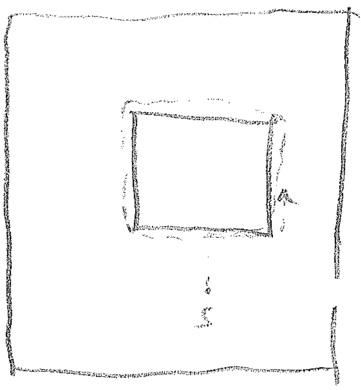
Shannon's mouse	Pledge
cellular environment, finite	polygonal environment, arbitrary angles, finite
touch sensor, able to 'follow' wall	— same here —
each cell is marked (E,S,W,N): mouse can alter marks	no marks
no memory	robot can sense turning a[ngle] and maintain angle coun[t]
mouse knows when target cell is reached (cheese!)	robot knows when it ha[s] escaped (sun is shining)

First approach

go north until wall is hit; ("north" = any direc[tion])
follow wall with left hand until escaped



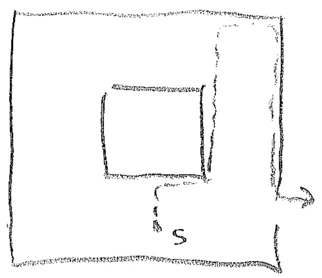
works



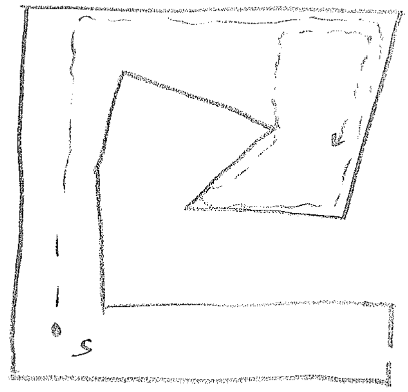
doesn't work

2nd approach

```
angle_counter := 0;
repeat
    go north until wall is hit;
    follow wall with left hand until
            angle_counter ≡ 0 mod 2π
until escaped
```



works



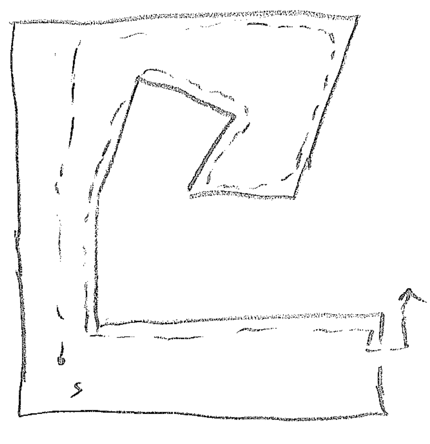
doesn't work

Pledge algorithm

```
angle_counter = 0;
repeat
    go north until wall is hit;
    follow wall with left hand until
            angle_counter = 0
until escaped
```

"free motion"

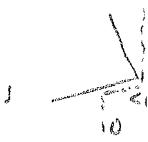
works in both example!



Theorem If there is a way out, Pledge will find it. (

Proof Indirect. Assume there is an environment P
from which Pledge does not escape

$\Rightarrow$ Pledge runs forever



Claim 0 At all times, angle counter ≤ 0. Proof: obvious

Claim 1 After initial phase, robot's path becomes cyclic.

Proof: Robot's path can turn only at one of the finitely many vertices of P.

case1 same vertex visited twice with identical angle counter $\Rightarrow$ path gets cyclic, since Pledge is deterministic

case2 no vertex visited twice with same angle coun

$\Rightarrow$ after some time, no vertex ever visited again with angle counter $= 0$

$\Rightarrow$ no more "free" movements between walls

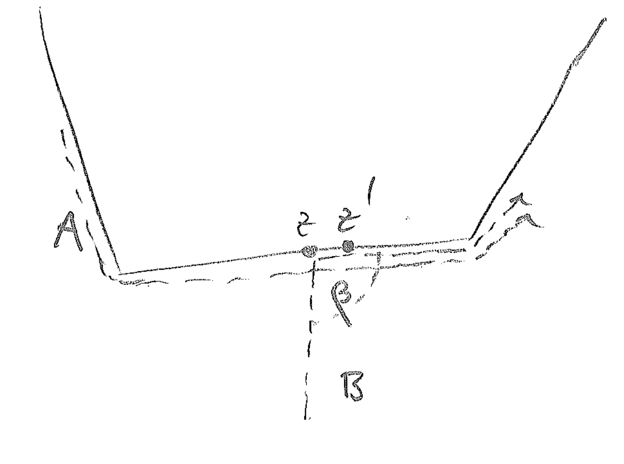
$\Rightarrow$ robot keeps following wall of single obstacle

$\Rightarrow$ path gets cyclic. $\square$

Let Z denote a minimal cycle in robot's path.

Claim 2 Z does not cross itself

Proof 2 Otherwise, Z contains segments A, B such that



AC(A), AC(B) := angle counter values at after coming from A resp. from B.

Clearly, $AC(B) = -\beta, \quad 0 \le \beta < \pi$

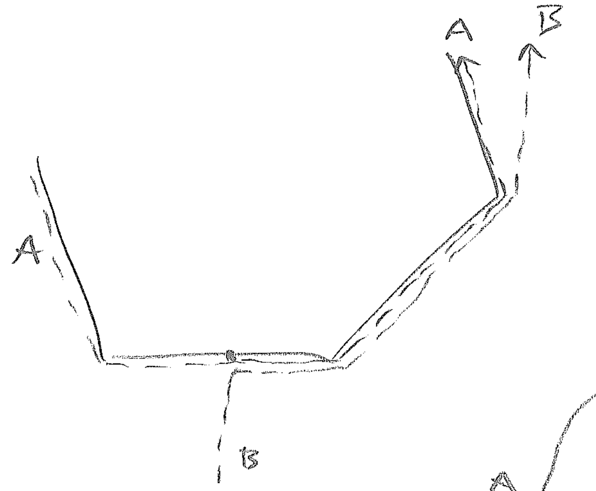
$AC(A) = -\beta + 2k\pi, \quad k \in \mathbb{Z}$

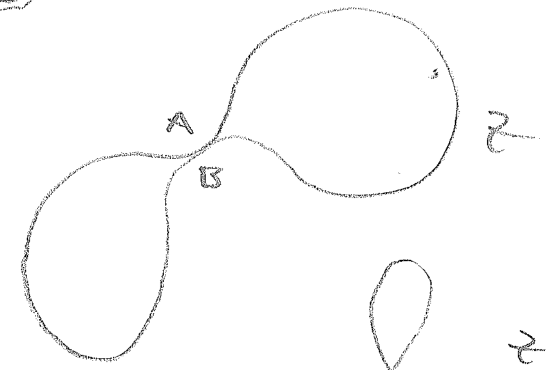
$k \geq 1 \Rightarrow AC(A) \geq -\beta + k\pi > 0$ ↯ Claim 0

$k = 0 \Rightarrow$ from z' on, A and B will run together.
Suppose segment A is visited next
$\Rightarrow$ B never visited again
$\Rightarrow$ ↯, A, B both segments of cycle Z

$k \leq -1 \Rightarrow AC(A) < AC(B)$
$\Rightarrow$ angle counter of B becomes 0 before angle counter of A does

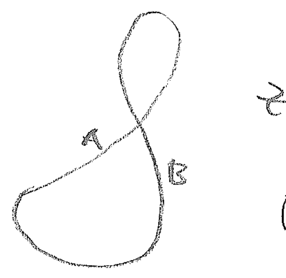
$\Rightarrow$





$\Rightarrow$ A, B touch

but don't cross



$\boxed{2}$

$\Rightarrow$ in each walk around cycle z,
angle counter (i) increases
or (ii) decreases } by 2π

(would not necessarily hold if crossings were possible)

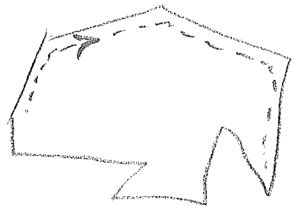
(i) increases $\Rightarrow$ after sufficiently many rounds, angle counter > 0 ↯ Claim 0

(ii) decreases $\Rightarrow$ after sufficiently many rounds, angle counter < 0 forever

angle counter: up and down during one round

$\Rightarrow$ no more free movements

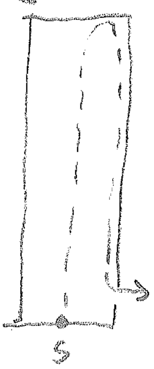
$\Rightarrow$

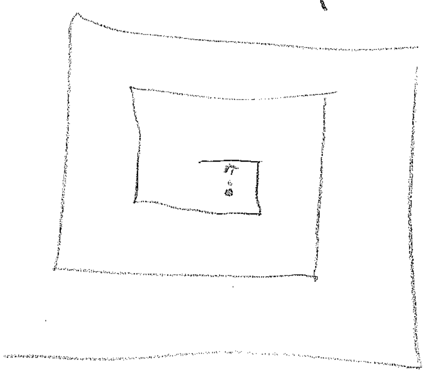


robot follows inside wall

$\Rightarrow$ no escape possible. $\boxed{\text{Theorem}}$

Pledge works, but may be inefficient:





<u>Change of model</u>:

robot wants to find target point, t
at each time, direction of t is known (compass)
no memory, no markers

Is it possible? Yes (Hemmerling, '94)
(in theory, not in practice)

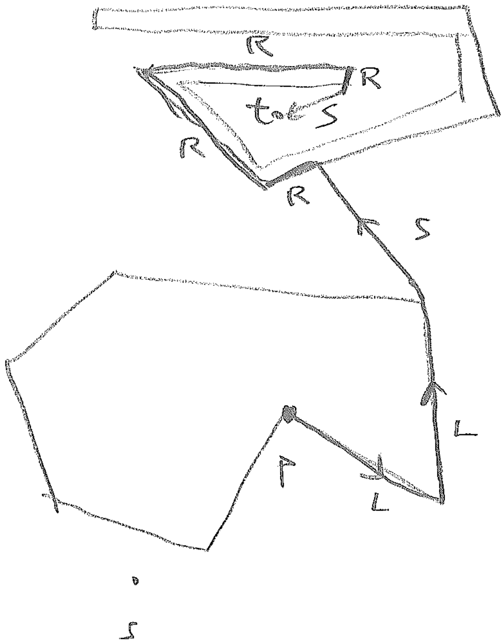
<u>Proof</u> Environment P has finitely many edges and vertices p.

robot's movements can be controlled by elementary commands

S : move straight to target t until t or wall gets hit

L : follow wall with left hand until vertex is reac

R : — right —



$w(p) = L^2 S R^4 S$ gets robo from p to t

Using the commands, robot can stop in

- vertices
- wall points hit by rays from vertices towards t
- s, t

Let $\{p_1, p_2, \dots, p_n\}$ be the set of all these points.

For each p_i, there exists word $w(p_i) \in \{S, L, R\}^*$ that guides robot from p_i to t.

Assume $w(p_1)$, applied to the robot at p_2, moves robot to p_m

$\Rightarrow$ $w(p_m)\,w(p_1)$ moves robot to t starting from p_1 <u>or</u> from p_2

$\Rightarrow \ldots \Rightarrow$ there exists $w \in \{S, L, R\}^*$ that moves robot to t no matter from which point in $\{p_1, \ldots, p_n\}$ the robot starts.

Problem: We don't know w.

But: We can enumerate all words $\in \{S, L, R\}^*$ (sorted by length / lexicographical order) and at some point, w will occur. $\square$

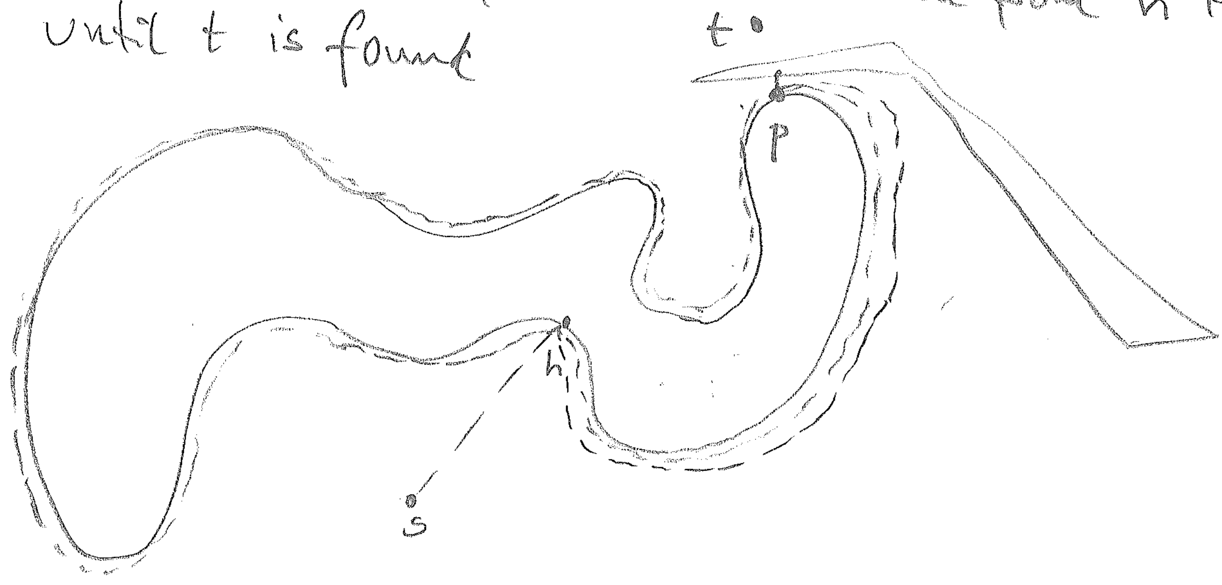
<u>More practical</u>: Lumelsky, Stepanov '87

<u>model</u> as before + robot knows distance from current position to target and length of paths traversed

($\Uparrow$ robot knows coordinates and has computing power)

BUG 1:

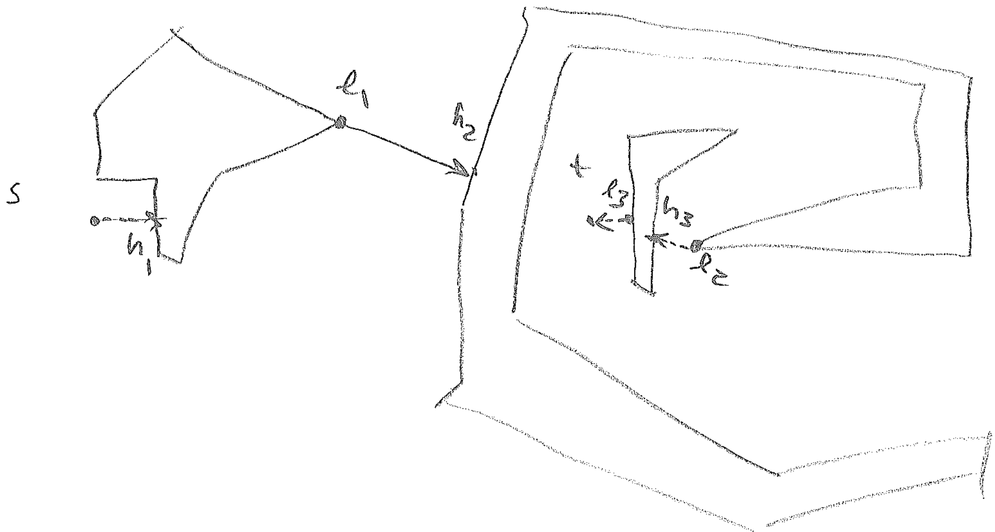
repeat
move towards t until obstacle wall is hit at point h;
walk around obstacle and determine point p closest to t
walk on shortest path around obstacle from h to p
until t is found



Theorem If t is reachable from s, BUG1 will find it on a path of length $\leq |st| + \frac{3}{2} \sum_{i=1}^{n} U_i$, where $U_1, \dots, U_n$ = perimeters of obstacles intersecting circle of radius $|st|$ centered at t.

Proof Assume obstacles are in general position (don't touch)

(h_i, l_i), $1 \leq i \leq k$ hit- and leave points



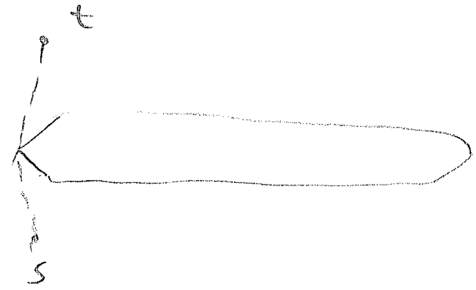
clear: $|l_i t| \leq |h_i t| < |l_{i-1} t| \quad \forall i$

$\Rightarrow$ each obstacle visited only once

$$|l_k t| \leq |h_k t| < |l_{k-1} t| \leq |h_{k-1} t| < |l_{k-2} t| \leq \dots \leq |st|$$

$\Rightarrow$ only such obstacles encountered that contain points satisfying $|pt| \leq |st|$ on their perimeters.
One full round, and at most half a round: $\frac{3}{2}$

Is BUG1 competitive? No!



But behaves quite well against other online strategies in this model!

Let S denote online search strategy in finite (polygonal) environment using touch sensor, target compass, distance measurement.

Theorem $\forall S$ $\forall D, \delta > 0$ small $\forall K > 0$ large $\exists$ environment E such that

$$|st\text{-path of } S| \geq D + \sum_i U_i - \delta$$
$$\geq K,$$

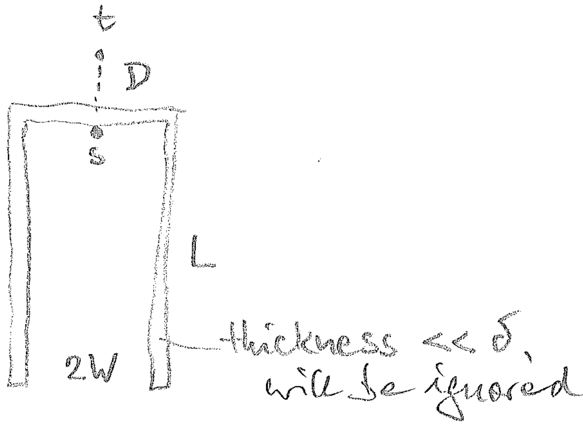
Here, $D = |st|$, U_i = perimeter of obstacles intersecting circle of radius $|st|$ centered at t

Remark Without K, Theorem would be trivial (Use empty environment)

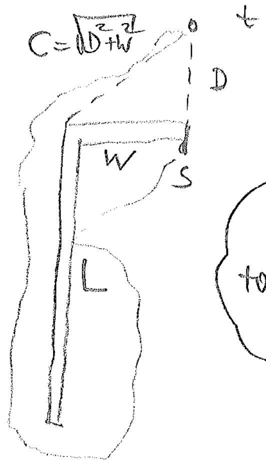
Proof Pick W large enough such that $\underbrace{D + W - \sqrt{D^2 + W^2}}_{\text{always } \geq 0} \leq \frac{\delta}{2}$

now, pick L large enough s.th. $L + W - \sqrt{L^2 + W^2} \leq \frac{\delta}{2}$

imagine (!) environment E:



Case 1 On its way from s to t, robot touches only left part,



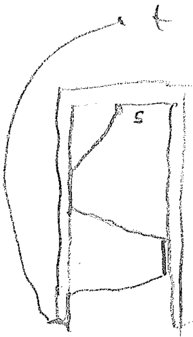
$$\Rightarrow \text{robot's path} \geq \sqrt{L^2+W^2} + L + \sqrt{D^2+W^2}$$

$$\geq L+W-\frac{\delta}{2} + L + D + W - \frac{\delta}{2}$$

$$= D + \underbrace{2(L+W)}_{} - \delta$$

$$= \sum_i u_i .$$

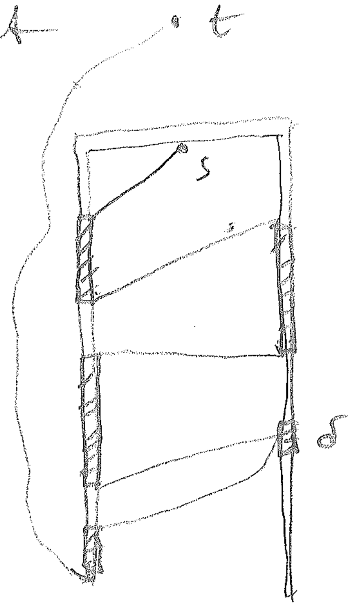
Define this to be environment E!



Case 2 Robot visits left and right part of

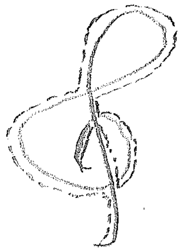
Can't argue path $\geq D + \underbrace{4(L+W)}_{\text{total perimeter}} - \delta$

Idea: Only wall segments touched by robot give rise to obstacles of E

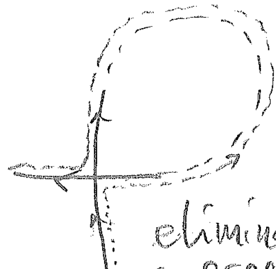


Let π denote robot's path (by S) may contain self-intersections

Lemma Each path in the plane can be traversed without crossings



Proof

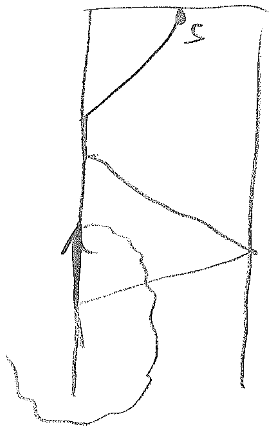


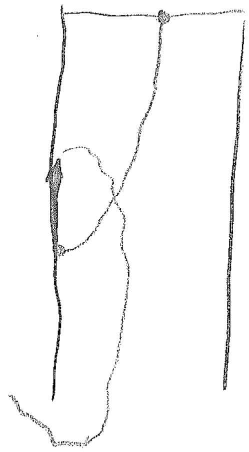
eliminates a crossing

Needs: Strategy produces paths w finitely many self-intersections

Once all crossings removed:
robot visits walls top-down, before leaving

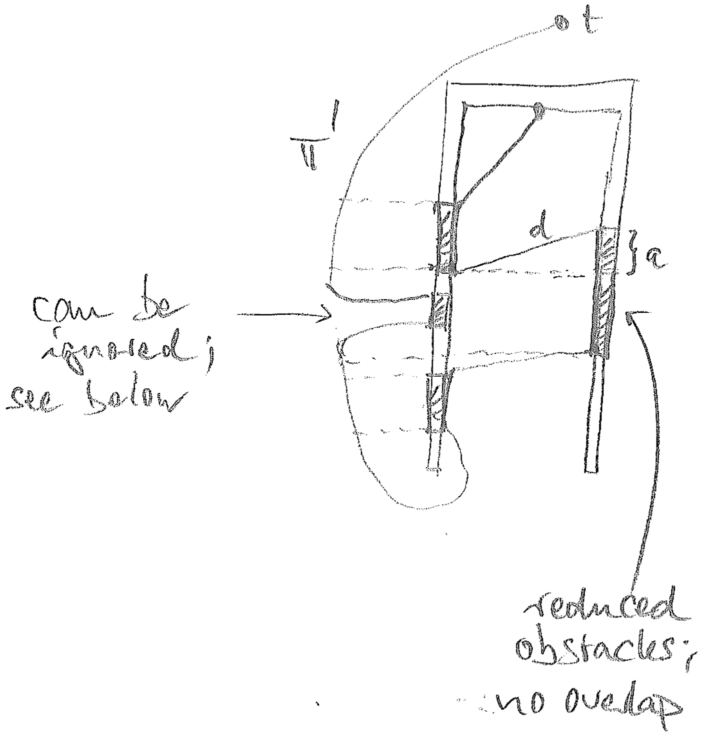
Otherwise,





crossing unavoidable.

Wall segments on opposite sides may overlap



Idea: $a < d$;

for each obstacle:
one edge visited by robot
the other edge
- charged to diagonal d, for overlapping part
- charged to path π' for non-overlapping segm

$$\text{robot's path} - \underset{\text{full obstacles}}{\sum_i U_i} \geq \underbrace{\sqrt{L^2+W^2}+L+\sqrt{D^2+W^2}}_{d+a \text{ forgotten}} - \underbrace{\underset{\text{reduced obsta}}{\sum_i U_i}}_{2a \text{ forgo}}$$

no more overlap
$$\geq \sqrt{L^2+W^2}+L+\sqrt{D^2+W^2} - \underbrace{2(L+W)}_{\text{all segmen beamed to le wall}}$$

$$\geq D-\delta \quad \text{as in Case 1.}$$

Theorems imply: BUG1 at most by a factor of $\frac{3}{2}$ worse than any other online search strategy (in $D + \sum U_i$ - terminology)

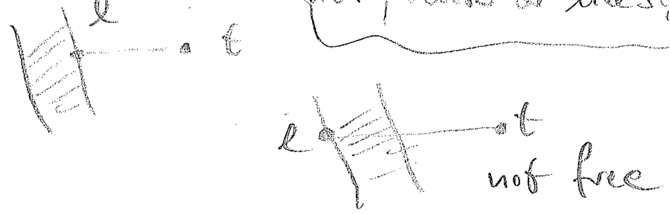
Variants

BUG2:

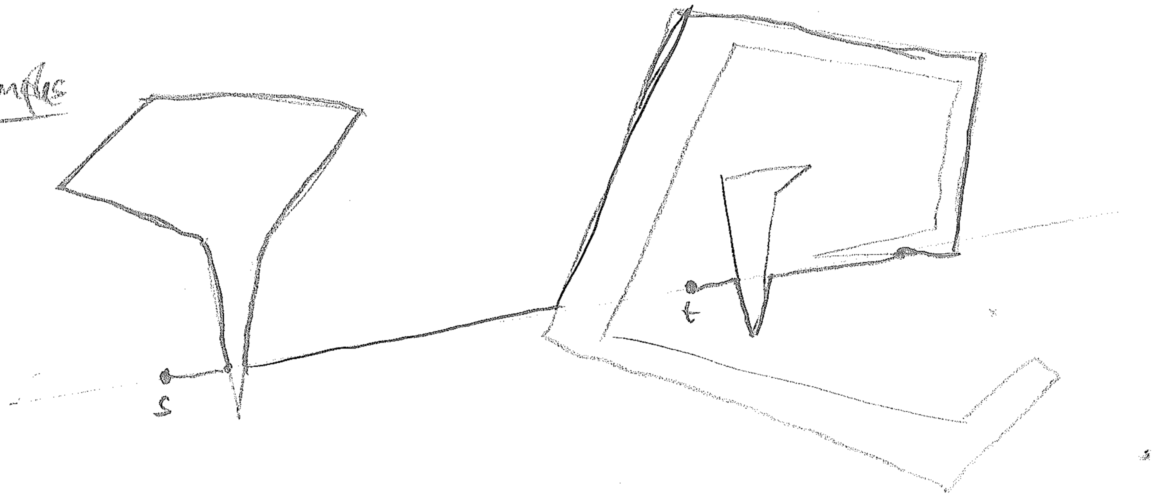
repeat
move towards t until obstacle wall is hit at point h
walk around obstacle (CW) until line $\overline{st}$ is hit
at some point ℓ where
- $|\ell t| < |ht|$
- $\overline{\ell t}$ is free

until t is found.

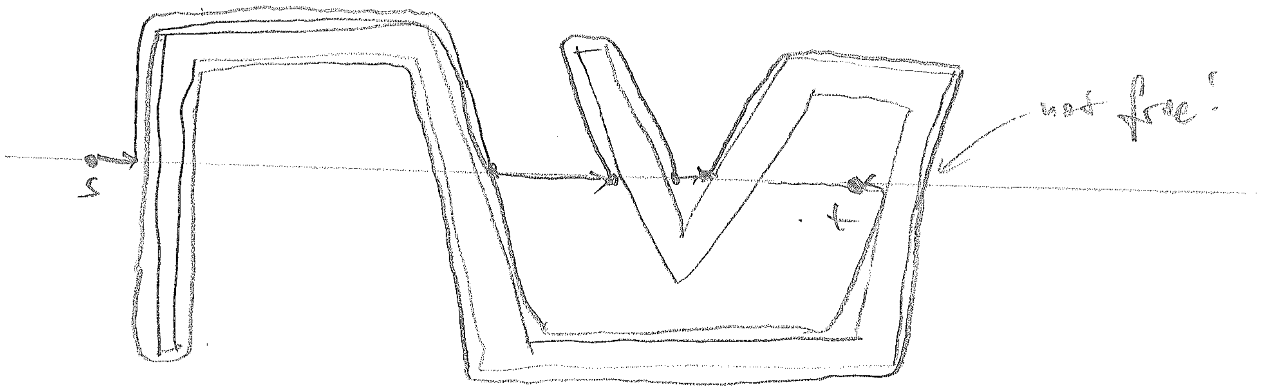
obstacles = polygons, not points or line segments



Example



looks much better - in this case. But



shows: same obstacle may be visited many times (not possible for BUG1!)

how often?

Lemma: Suppose line(s,t) intersects boundary of obstacle P_i in n_i points. Then each point on the boundary of P_i is visited at most $\frac{n_i}{2}$ times.

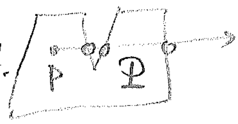
("general position" ⇒

does not happen

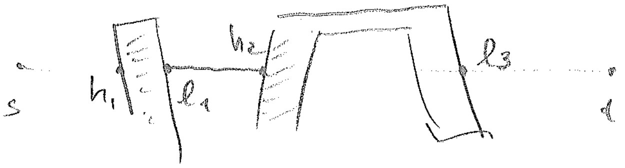
here: disappears after perturbatio

Proof: Observe that n_i is even (can be used for inside / outside test

3 points of intersect. ⇒ p lies in P



Consider sequence $(h_j, l_j)_j$ of hit-/and leave points generated by BUG 2



Clear: $|h_j t| > |l_j t| > |h_{j+1} t|$

⇒ each boundary point on li can at most once be a hit and only $\frac{n_i}{2}$ candidates ex

$\boxed{L}$

Theorem If t is reachable from s then BUG2 finds a path of length

$$|\text{path}| \le D + \sum_i \frac{n_i}{2} U_i , \qquad \text{where } D = |st|$$

and U_i = perimeters of obstacles intersecting circ$(t, |st|)$

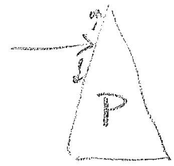
$n_i = |\text{line}(s,t) \cap \partial P_i|$.

Proof By previous lemma. Each obstacle boundary can be traversed at most $\frac{n_i}{2}$ times. $\boxed{Th}$

Corollary: If all obstacles are convex, BUG 2 is "optimal" among all on-line strategies

(in the sense of lower bound theorem, not in the sense of rigid competitive analysis.

Problem for BUG1 and 2:
will consider to approaches.



go cw or ccw
around ∂P?

Classical problem: door in a wall

Hier Einschub: Tür in Wand
P0 – P5.2 aus Vorlesung
"online Algorithmen"