# Online Motion Planning MA-INF 1314

## Example questions

Elmar Langetepe

Universität Bonn, Institut für Informatik

19.7.2016

# Questions

1. CFS Algorithm
   Example in detail
2. STC-Algorithms
   Scan-Variant
3. Online Navigation
   Lower bound construction

# General: For any configuration

1. Precise model
2. Strategy/Algorithm
3. Correctness
4. Lower Bound
5. Upper Bound
6. Structural properties, proofs
7. Main statements
8. Extensions, Applications, Remarks

# General: Lemmata/Theorems

1. Top/Down
2. Main statements
3. How did we achieve them?
4. Lemmata
5. Structural properties

# SmartDFS

1. Definition: Grid-environment, online, vertices, simple, neighboring cells
2. Theorem: Number of steps C $+1/2$ E -3
3. Strategy Idea: SmartDFS, split-cell, recursion, quadrant Q by layer,
4. L-Offset-Lemma: $8l$ edges less!
5. Shortest-Path-Lemma: $1/2E - 2$
6. Edge-Lemma for $Q$: $E(P_1) + E(P_2) = E(P) + E(Q)$
7. Excess-Lemma: $excess(P) \leq excess(P_1) + excess(K_2 \cup \{c\})$
8. Proof Theorem: Induction! Apply Lemmata!

# General: For any configuration

1. Precise model
2. Strategy/Algorithm
3. Correctness
4. Lower Bound
5. Upper Bound
6. Structural properties, proofs
7. Main statements
8. Extensions, Applications, Remarks

# CFS Algorithm

1. Precise model: Constrained graph exploration, edges and vertices, Tether variant vs. Accumulator variant, depth restriction variant
2. Strategy/Algorithm: CFS, example application
3. Correctness: By construction
4. Lower Bound: $E + V$
5. Upper Bound:: $(4 + 8/\alpha)E$ (more precisely: $\Theta(E + V/\alpha)$)
6. Structural properties and poofs: Proof of the invariants, analysis of the cost
7. Main statements: Competitive online exploration, UB, depth restricted, lookahead
8. Extensions, Applications, Remarks: Search ratio approximation, adjustments for unknown depth, simulate accu-variant by thether-variant, lookahead is necessary (accu)

# STC Algorithms

1. Precise model: Exploration, 2D cell, visit all cells by the tool, scan the 4 neighbourship of 2D cells. Return variant!
2. Strategy/Algorithm: Spanning Tree construction online (DFS), Tool left hand side
3. Correctness: Visit all 2D cells that can be entered
4. Lower Bound: Visit all cells C (Hamiltonian path!)
5. Upper Bound: C + K (Boundary cells)
6. Structural properties, proofs: Example execution and analysis! Inner/Intra double visits
7. Main statements: C + K, 2-competitive, optimal in pure 2D scenes, Tightness for corridors
8. Extensions, Applications, Remarks: Scan-STC, avoid spanning tree edges of a special kind, simple heuristic/analysis: $H_{opt}$ + Column-Divergence +1

# Online Navigation

1. Precise model: Touch sensor, coordinates of the goal, two different movements
2. Strategy/Algorithm: BUG variants, example executions, intention
3. Correctness: Closer to the goal, leave condition, enclosed?
4. Lower Bound: Distance to the goal, plus circumference of the the obstacles
5. Upper Bound: Depending on the variant
6. Structural properties and proofs: BUG2, intersections, tight bounds, estimating the movement in the free space, LB construction circumference
7. Main statements: Correctness, robust strategies, performance
8. Extensions, Applications, Remarks: Change1, Change2, Visibility