Examples of possible questions in oral examinations

Part Search

Q 1: What are nodes and edges representing in an OR graph?
Q 2: What is a solution path in an OR-graph?
Q 3: What are constraint satisfaction problems? What are optimization problems?
Q 4: What is an appropriate representation for infinite graphs?
Q 5: What is node expansion?
Q 6: What are locally finite graphs? Why do we need them?
Q 7: What is a solution base (differences to solution paths)?
Q 8: What is an efficient way of representing solution bases?
Q 9: What is the tree Basic-OR-Search maintains?
Q 10: Why is the graph maintained by Basic-OR-Search a tree?
Q 11: Why is the traversal tree in Basic-OR-Search no subgraph of the OR-graph?
Q 12: Are DFS and BFS variants of Basic-OR-Search? Why? / Why not?
Q 13: Comparison of DFS and BFS: Which algorithm is to be preferred when and why?
Q 14: Which nodes are stored on OPEN, which nodes on CLOSED?
Q 15: Why is a function *cleanup_closed*() needed in DFS?
Q 16: What is iterative deepening?
Q 17: What information sources does the evaluation function f in BF use?
Q 18: What are the main differences between UCS and BF?
Q 19: What is the difference in the evaluation functions of UCS and BF?
Q 20: What is path discarding?
Q 21: When using path discarding, is the traversal tree a subgraph of the search space graph?
Q 22: Why can path discarding be problematic?
Q 23: What does node reopening mean?
Q 24: What is $C_P(s), C^*(s), \widehat{C}_P(s), \widehat{C}(s)$?
Q 25: How do we define an evaluation function f by a cost function $\widehat{C}$?
Q 26: How do we define recursive cost functions?
Q 27: How can we define a function $\widehat{C}_P(n)$ for estimated solution cost on basis of recursive cost functions?
Q 28: Why can it be an advantage to use recursive cost functions?
Q 29: Can cost functions help to avoid problems in path discarding?
Q 30: What is the evaluation function used in algorithm A*?
Q 31: What is h and what is g in the evaluation function of algorithm A*?
Q 32: What is path cost in algorithm A*?
Q 33: Is the underlying path cost function $\widehat{C}_P(s)$ in A* order preserving? Is this only true if we have negative edge cost values?
Q 34: Why do we need delayed termination in order to solve optimization problem? (Example?)
Q 35: What is an optimistic evaluation function?
Q 36: Why do we need optimistic evaluation functions in order to solve optimization problems? (Example?)
Q 37: What is the motivation for specifying *Prop*(G) for search space graphs?
Q 38: What is the consequence of a positive lower bound of edge cost values for long paths?
Q 39: Is existence of optimum cost solution paths guaranteed for search space graphs with *Prop*(G)?
Q 40: What is completeness, what is admissibility for search algorithms?
Q 41: What are the main steps in proving completeness of A*?
Q 42: Why can't we prove termination of A* on infinite graphs?
Q 43: What is a shallowest OPEN node?
Q 44: How do shallowest OPEN help proving completeness?
Q 45: What is the additional property of shallowest OPEN nodes on optimum cost paths that is used for proving admissibility of A*?

Q 46: What is the statement of the C* bounded OPEN node lemma?
Q 47: What is the definition of an admissible heuristic function?
Q 48: What is the idea of the proof of the C* bounded OPEN node lemma?
Q 49: What is the statement of the admissibility theorem for A*?
Q 50: If we use a solution path $P_{s-\gamma}$ with cost $C \geq C^*$ instead of an optimum cost solution path, what is the statement we can prove instead of the C* bounded OPEN node lemma?
Q 51: What necessary and sufficient conditions for node expansion by A* did we consider?
Q 52: What are the nodes considered in necessary and sufficient conditions for node expansion by A*?
Q 53: How can we increase efficiency by applying the necessary condition for node expansion of OPEN nodes by A*?
Q 54: How is monotonicity (consistency) for heuristic functions defined?
Q 55: How can monotonicity be proven from consistency? (Proof ideas.)
Q 56: How can consistency be proven from monotonicity? (Proof.)
Q 57: Why is it important to have both, monotonicity and consistency?
Q 58: Are monotone heuristic functions admissible? (Proof.)
Q 59: Consider the 8-puzzle. Give an example of a monotone heuristic function.
Q 60: What is the advantage of using monotone heuristic functions in A*?
Q 61: Give the outline of the proof of the No Reopening Theorem.
Q 62: Should we always prefer monotone heuristic functions over admissible ones?
Q 63: If we have two heuristic functions, the one more informed than the other on part A of the search space graph and the other way round on part B, which heuristic function should we use in A* search?
Q 64: Why is solving optimization problems with A* search an efficiency nightmare?
Q 65: What is the idea of the weighing approach?
Q 66: Why do we expect that the search effort in WA* less than in A*?
Q 67: What properties should h have in WA*, what properties should $(1 + \varepsilon)h$ have?
Q 68: What is the idea of the A*$_\varepsilon$ algorithm?
Q 69: What properties should h have in A*$_\varepsilon$, what properties should h_F have?
Q 70: Why do we expect that the search effort in A*$_\varepsilon$ less than in A*?
Q 71: What are the differences of WA* and A*$_\varepsilon$ to A* in pseudocode?
Q 72: What cost C of a solution path can be expected for WA* and A*$_\varepsilon$? (Preconditions.)
Q 73: Which of the two algorithms WA* and A*$_\varepsilon$ is more powerful? (Using appropriate functions h and h_F.)
Q 74: What is the essential step in proving that A*$_\varepsilon$ can simulate to WA*?
Q 75: Is WA* complete? Is A*$_\varepsilon$ complete?
Q 76: How can we prove ε-admissibility of WA*?
Q 77: How can we prove ε-admissibility of A*$_\varepsilon$?
Q 78: Does A*$_\varepsilon$ benefit from using monotone heuristic functions in the same way A* does?
Q 79: What is restricted path discarding?
Q 80: What is the relation of cost C of a solution path found by NRA*$_\varepsilon$ to C^*?

Part Planning

Q 81: What is a STRIPS model and a STRIPS planning problem? Explain states, goals, CWA DC, UNA.
Q 82: How do we get from operators to actions?
Q 83: What is a (solution) plan?
Q 84: What is state-space planning?
Q 85: How does regression work?
Q 86: How can we construct heuristic functions for A* search in forward planning?
Q 87: Explain the partial-order planning approach.
Q 88: What is the objective in HTN planning (compared to state-space planning, plan-space planning) ?
Q 89: What is a task network?

Q 90: What is the main step in HTN planning?
Q 91: How does total-order STN planning work?