Instuderingsfrågor -- av Rasmus Johns

1. What is a agent, agent function and agent program? (1p) ()

Agent: Something that perceive its world and acts upon its perception through actuators.
Agent function: An abstract mathematical description of the agents behaviour. It maps input to output (more formally : Maps every percepts sequence into an action f : P* -> A)
Agent program: An implementation of the agent function.

2. What is the performance measure, rationality and autonomy of an agent? (1p)

Performance measure: A measure for how the agent is performing. “How well can the agent do it’s mission?”
PEAS:

Agent Type	**Performance Measure**	**Environment**	**Actuators(actions)**	**Sensors**
Vacuum cleanah	Use minimal power Suck up all the dirt etc	Apartment w/ kitchen, living room, trösklar	Suckisucki, go straight, turn 90 degrees	THAR IS DIRT And bump

Rationality: “Does the right thing at the right moment”. Based on what the agent knows about the world, it’s percepts and actions, it tries to maximize it’s performance with every action.
Autonomy: An autonom agent acts upon it’s own perception, and not the experience of it’s manufacturer.

3. Describe a reflex agent using text and illustrations. (1p)

Environment, Agent (Sensors -> What does the world look like + conditional rules -> action -> actuators)
Acts entirely based on perception, keeps no model of the world. Ot only knows the world it sees and maps current input to output.

4. Describe a model based agent using text and illustrations. (1p)

Add *State*, *How the world evolves* and *What does my actions do* to the reflex-based.
This agent keeps a model of the world (internal memory!), where it can save what it sees and how the world works.

5. Describe a goal based agent using text and illustrations. (1p)

Add *Future worlds* / *What will it be like if I do action A* as well as new arrows from state, evolve and what actions do. Change rules into goals.
This agent have goals instead of rules. It tries to find a sequence of actions which will place it in a goal state.

6. What is PSS? (2p)

A physical symbol system (also called a formal system) takes physical patterns (symbols), combining them into structures (expressions) and manipulating them (using processes) to produce new expressions.

7. What is the PSS hypothesis? (1p)

"A physical symbol system has the necessary and sufficient means for general intelligent action."
Necessary: Any system that exhibits general intelligence needs to manipulate/interpret symbols in some way.
Sufficient: Any, large enough, PSS can be organized to further to exhibit general intelligence.
This claim implies both that human thinking is a kind of symbol manipulation (because a symbol system is necessary for intelligence) and that machines can be intelligent (because a symbol system is sufficient for intelligence.

8. Can a PSS act intelligently? (2p)

2p för att skriva ja/nej/typ? Wow.

Asså vem är jag som ska komma här o säga att Newell & Simon hade fel?

9. What is supervised learning? (1p)

Give the learning agent a set of correct input-output pairs that it's supposed to be able to handle. The agent tries to find the function f(input) = output given the pairs of (input, output). "SL mimic true function."

10. What is reinforced learning? (1p)

"RL looks at past actions and outcome." An agent is given reward for based on it's performance. The agent tries to find what action generates the highest reward.

11. Describe the different parts of Q-learning. (2p)

$Q(s, a) \leftarrow Q(s, a) + \alpha(R(s) + \gamma \max Q(s', a') - Q(s, a))$
Learning rate (alpha): A higher value makes the agent "give greater significance" to given reward and less to previous reward. If it equals one though the world better be deterministic and discrete, because then Q(s,a) will be canceled out and then only given reward is stored.
Discount factor (gamma): Value between 0 and 1. A high gamma makes the agent more patient while a low value for gamma makes it greedy and short-sighted. Gamma is used to make (the utility function) the sum of accumulated rewards finite, even for infinite sequences.
Q-table Q(s,a): Table of estimated utilities for taking action a in state s. Updated with each action the agent perfom, values is updated based on observed value and observed next state.
Reward R(s): A value the agent receives dependent on the state. Could be a positive reward for i.e. winning a game or a punishment for losing.

12. Is adding more states and actions to Q-learning beneficial?

That depends! The Q-table grows exponentially with the action-state-space so more states/actions could make the Q-learning be very inefficient (taking a long time to converge) or not functional at all.

13. Why is exploring needed in reinforced learning? (1p)

Without exploring the agent is very likely to get stuck in a suboptimal behaviour, because it doesn't explore its world thoroughly.

14. Describe the MRV-heuristic. (1p)

Minimum Remaining Values: Choose the variable with the smallest domain (The variable with the fewest remaining legal values it can be assigned to).

15. Describe the degree-heuristic. (1p)

Choose the variable with the highest degree - the variable which is involved in the largest number of constraints on other unassigned variables.

16. Describe the least constraining value heuristic. (1p)

While 14 & 15 are two variable selection strategies least constraining value heuristic is about what value to assign a variable to. Choose a value for a variable such that other the value domains of other unassigned variables are affected as little as possible.

17. What is forward-checking? (1p)

Look at how other variables are affected by an assignment: When assigning a value to a variable X, update the domain of those variables connected to X by a constraint. The "update" refers to removing invalid values from said domains. "Given the current partial solution and a candidate assignment to evaluate, it checks whether another variable can take a consistent value". Backtrack if a value domain becomes empty.

18. What is a satisfying plan? (1p)

That would be a plan which isn't necessarily the best given the problem - we're just looking for good enough solution.

19. What is relaxed solution?

A relaxed solution is a solution to a relaxed problem, which in turn is explained in 26.

20. (Question about planning) Describe "open goals" and "threats". (2p)

Those are so called flaws. Open goals is goals we want to achieve (unlinked or not achieved by any action). Open goals are resolved by either
1. adding a causal link to it, from an action which achieves the goal
2. adding a new action which has the open goal as an effect.
Threats is states achieved of some action in our planning which makes a following action illegal/invalid.
Threats is contradictions in the planning. Say that initially clear(A) holds. One action has this as a precondition but another parallel action makes it so not clear(A) holds. Hence - we can't be certain that clear(A) holds when the action needing this to be true is to be carried out. This conundrum is resolved either by
1. Make the actions appear as a sequence in the planning. First do the action with precondition clear(A) and then follow that action up with the other one.
2. Place the action disturbs the precondition before the action supporting it.

21. Describe arc consistency. (1p)

Given two nodes Vi and Vj, connected by a constraint (an arc). If for each value x in the domain of Vi there exist some value y in the domain of Vj such that Vi = x, Vj = y is a legal assignment, the arc is consistent.

22. What is an admissible heuristic? (1p)

That would be a heuristic that never overestimates the cost of completing a search/problem from a given node/state.
A heuristic h(n) is said to be **consistent** if for every node N and for each successor P of N, the estimated cost of reaching the goal from N is no greater than the step cost (c(N,P)) of getting to P plus the estimated cost of reaching the goal from P. That is:
$h(N) <= c(N,P) + h(P)$ and
$h(G)=0$
Alternative way of describing it: h(n) - heuristic. c(n, a, p) - the cost of going from n to p through arc a.
$h(n) \leq c(n, a, p) + h(p)$.
It's important that the used heuristic is admissible if one would want to acquire the optimal solution to a problem. A* proof for optimality when h(n) is admissible:
Say that A* has two current nodes in the frontier, the suboptimal goal G2 and a node P on the actual optimal path. The cost of the optimal solution to this is C*.
A* will choose the node with the lowest calculated cost according to f(n) = g(n) + h(n) where g(n) is the cost of going from initial node to node n. The cost of f(G2) = g(G2) + h(G2) = g(G2). This must be of a larger value than C* since C* is the optimal solution and G2 is indeed suboptimal.
$C^* < f(G2)$!
The node P on the other hand is on a optimal solution path. If h(n) never overestimates then $f(P) = g(P) + h(P) \leq C^*$
Since $f(P) \leq C^* < f(G2)$ will P get expanded and G2 ignored, for now. And you know what??? A* is optimal indeed!

23. Given a book of infinite answers to any sequence of symbols possible, can this book be seen as intelligent? (1p)

If the book itself can look up a correct answer to any given sequence, then sure. INFINITE WISDOOOOM!

24. Describe the Turing test. (2p)

So there is this guy, I, who tries to figure out whom is computer C and whom is the human H on the other side of some intermediary. I asks questions, H is all human-like and… Wait what's this? So is C!!! What is the interpreter to do?? If C can fool I into believing C is in fact the human as many times as it fails in doing so - then my my, this computer is something.
The test, "The Imitation Game", described above was according to Turing a more meaningful way to ponder about the question if "machines could think", since it offers a much more precise question: "Could a computer do well in this game?".

25. Would you say argue that the Turing test measures a computer's intelligence? (1p)

If by intelligence you mean ordbajs-skills, then yes.

Nedanstående tagna från FÖ om planning. "Describe and explain:"

26. Relaxation, and how it can be applied to create an admissible heuristic

HAHA NOT YET IS ISNT

27. PDB heuristics

Pattern Database heuristics.

28. Landmarks

29. The main advantage of partial-order planning over forward state space planning

30. The different kinds of flaws that can occur in a partially ordered plan

31. The ways in which can you resolve an open goal

32. Why planners don't use Dijkstra's algorithm to search the state space

33. Is the number of open goals an admissible heuristic? Why / why not?

34. Given a domain, problem and initial state: Expand the forward state space by two levels

35. How can you apply hill climbing to planning?

36. How do you have to modify standard hill climbing, and why?

Gamla tentasvar (Tenta för 2015-08-24 finns även i ett doc):

https://drive.google.com/drive/folders/0B2ylu4i-VKa-T2IxLTFtS0h0RVU?usp=sharing