AI: Cheat sheet Q1: all qs refference cheet sheet on GH!

- Always talks about an agent
- agent: acts on an enviroment
- intelligent agent: (†)
  - acts intelligently (duh)
  - The actions it takes are appropriate for it's goal/achi
  - learns from it's experiences
  - makes appropriate choices given perceptual limits and finite computation
- symbol-system: ~~is the~~ Reasoning is symbol manipulati (the necessary and sufficient means for general intelligent actions)
- The question: 1(a) (†) is an intelligent agent, makes appropriate choices given .... and the symbol-system is any system that acts intelligently ∴ our agent follows the SSH!
- Non determinism: is an algorithm type that, given the same input, can have a completely different outcome/ behaviors.
- (b): when (†), an intelligent agent, acts intelligently on a search algorithm, it must make certain choices during it's computation. This is why when given the same input for the search, it can ultimately give us a different answer as it has acted intelligently upon the algorithm (guessed choices). This is why the search algorithm ~~relates~~ to non-determinism!

- Cantor's theorem: is a fundamental result that states that for any set, the set of all subsets of (the power set of, denoted by) has a strictly greater cardinality than itself!

(c): (state cantor's). The power set of an an countably finite set, is uncountably infite as it consists of the cardinality of the real numbers.
∴ To search an infitite bit strings, would mean you would have to search it's power set that is uncountably infinite.

- SAT problem: is the problem of determining if there exists an interpretation that ~~sa~~ satisfies a given Bodeen Formula

(d): (state SAT): It relates to the finite bit string as it is either found or not found through intelligent choices by the agent (not too sure on this one)..

(e): (state SAT). With n boolean variables there are 2^n different possible bit strings of length n. This makes the search space exponential. In principle it can be searched but is very expensive. ~~As it~~ To search this space using brute force takes worse than polynomial time.

(j) (state sect). The agent might be trying to find an assignment to the variables satisfying an expression. (since it intelligently uses it's inputs) Boolean expressions are a way of expressing what it's trying to complete.

P vs NP asks whether every problem whose solution can be quickly verified can also be solved quickly.

(k) (stat P vs NP). SAT can then be feasiable if P=NP as N allows for non-determinism (ie the agent takes control of the outcome).

Church-Turing thesis: states that any symbol manipulation can be carried out on a Turing machine

(L) (state CTT). our agent is known to act intelligently and reinforces the fact that it can (as a TM) through the intelligent manipulation of symbols. (?)

Constraint satisfaction problem: is a mathematical problem that is defined as a set of objects whose state must satisfy a number of constraints/limitations

Var = [$x_1, \ldots x_n$] of variables x_i
Dom = [$D_1 \ldots D_n$] of finite sets D_i of size s_i

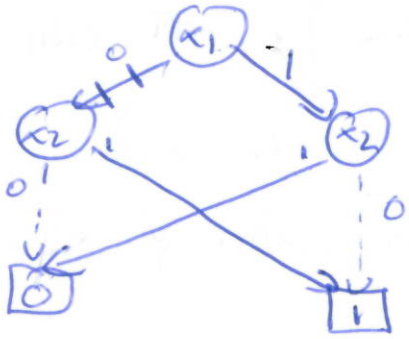
Con = a finite set of constraints that may/may not be satisfied by (a node) instantiantiating x_i with a value in D_i

Binary decision diagram (BDD): Answer to (f)

It is a data structure that is used to represent a boolean function.

ordered: if different variables appear in the same order on all paths from the root

reduced: when low/high children of the node cannot be the same



ROBDD

$(x_1 \text{ or } x_2)$ and $(\bar{x_1} \text{ or } \bar{x_2})$

Answer to g

A BDD is satisfiable when it completes the SAT problem.

Turing Machine: is an abstract machine that manipulates on a tape of symbols using a table of rules.

(h) (state turing machine). (f) relates to one as it can be modelled as a TM where tape is the enviroment.

Halting problem: on a turing machine (computer program), determines (given an input) if it should halt or continue. $h(i,x) = \begin{cases} 1 & \text{if program } i \text{ halts on input } x \\ 0 & \text{otherwise} \end{cases}$

(i) (state halting problem). It is impossible to determine if (f) will ever find a solution!

Eg- SAT = Di {0, 1}, si = 2 for search space size 2^n

4) state var, Dom, con

5) SAT approach above (I think!)

6) ~~A* start~~ For an algorithm to be admissable it must not overestimate the cost of reaching the goal. (current point estimated ~~must~~ cost must not be > than the lowest possible cost to that node).

3 conditions:

must not underestimate

termination; For some $\epsilon > 0$, every arc costs $\geq \epsilon$ (estimation)

finite branching {n' | arc(n, n')} is finite for each node n

7) see BDD

8) Non-determinism can be applied to search with regards to a*. It uses the heuristic function to determine the shortest path, but can sometimes give us different outputs with the same inputs! The (+) uses this to calculate the search intelligently!