2016

April 29, 2017

Question 1

a

A physical symbol system takes physical patterns (symbols), combines them into structures (expressions), and manipulates them to produce new expressions.

An agent acting intelligently can be represented as a physical symbol system. The environment can be represented by symbols, and the action can be represented as symbol manipulation.

b

A nondeterministic algorithm is one that, even for the same input, can exhibit different behaviours on different runs.

An agent acting intelligently must exhibit some nondeterminism, as intelligent action requires making choices. Paradox of AI is that computation is deterministic.

Searching can be computed nondeterministically (think of a man walking down a path in a forst and every time he steps further, he must pick which fork in the road he wishes to take). These algorithms are guaranteed to arrive at a correct solution for some path. The choices are guesses in the search process.

c

Cantor's theorem is a fundamental result that states that for any set A, the power set of A has strictly greater cardinality than A itself. The power of set of an countably infinite set (a set with cardinality the same as the natural numbers), is uncountably infinite and has the same cardinality as the real numbers.

These means that in a set of infinite bit strings, to search these strings would mean to search the power set of infinite bit strings, which is said to be uncountably infinite.

d

The problem os SAT of Boolean satisfiability is of determining if there exists an interpretation that satisfies a given Boolean formula. With n boolean variables there are 2^n different possible bit strings of length n, i.e. the search space is exponential. To search this space using a brute force solution takes worse than polynomial time, but there may be more efficient ways to solve the problem.

e

A BDD is a structure used to represent a Boolean function. A BDD is ordered when variables appear in the same order along all paths from the root to the leaves. A BDD is reduced when low and high chilcren of the node cannot be the same.

(x1 or x2) and (¬x1 or ¬x2)

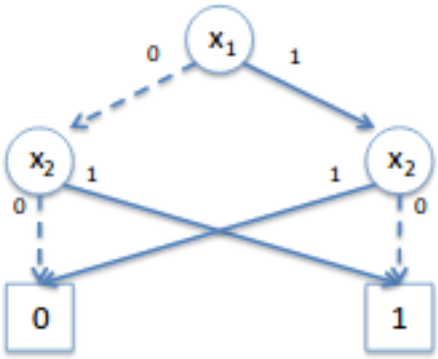

Figure 1: ROBDD

A BDD is satisfiable when it is not equal to 0, i.e. there is some path so that the value is 1.

Question 2

a

```
goalL([H|T]) :- goal(H); goalL(T).
arcL(List, NextList) :- findAll(X, (member(Node, List), arc(Node, X)), NextList).
```

b

A heuristic function estimates the cost of the shortest path from node n to a goal node, using readily obtainable information about the node. The frontier is then organised as a priority queue of the lowest h-value.

- The `Node` would need to be a `(Node, Value)` pair where `Value` is the value of the heuristic function.
- Currently, `add2frontier` gets the current list of nodes to be search and appends a list of newly found nodes to be searched (based on head of the current list). This would need to be altered so that the frontier is re-arranged for every newly found set of nodes to be searched, ordered by heuristic value.

c

A-star not only has a frontier which is a priority queue, and uses a heuristic function, but it is ordered by the heuristic function *and* the cost to get from the current node to the next node.

$$f(n) = cost(n) + h(n)$$

I.e., it is the total path cost of going from the current node to a goal node via the next node.

A-star is admissible if it will find a solution if there is one, and it will have the minimal cost value. This happens if

- the branching factor is finite
- arc costs are bounded above zero
- the heuristic function is an underestimate of the length of the shorest path from a node to a goal node.

d

i

Length of $[X_1, \ldots, X_n]$ is $s_i \times n$

The length of the list to test becomes the product of the subset lengths $D_1, \ldots, D_n$

ii

- Depth: N
- Branching factor: the length of the largest subset

iii

If you instantiate more than one variable you're overestimating. Under these conditions, best-first search becomes breadth first search.

Question 3

a

Two nodes of the same colour cannot be beside each other.

(R, R) false	(R, G) true	(R, B) true
(G, R) true	(G, G) false	(G, B) true
(B, R) true	(B, G) true	(B, B) false

b

C is a logical consequence of KB if C is true in every model of KB.

C is satisfiable if it has a model. This is not necessary the same as being a logical consequence, as C must be true in every model for it to be a logical consequence.

c

If a knowledge base is sound, every atom g that a proof procedure derives follows logically from KB, i.e. $KB \vdash g$ implies $KB \vDash g$

If a knowledge base if complete, every atom g that logically follows KB is derives, i.e. $KB \vDash g$ implies $KB \vdash g$

$KB \vdash false$ means $false$ can be derived from KB, this implies there is no interpretation where all clauses in KB are true. I.e. `i` implies `ii`.

There is no interpretation where all clauses in KB are true, this implies there is a clause where KB is false, so false can be derives from KB ($KB \vdash false$), i.e. `ii` implies `i`.

d

- `{r}` is a conflict because if `r` is true then `a` is true which is a conflict of the rule `false :- a`
- `{r, p}` is a conflict, although not a minimal one as `r` is a minimal conflict

e

A reasoning system is non-monotonic is a conclusion can be invaidated by adding more clauses.

Yes, as false can be invalidated if q is true.

Negation as failure leads to non-monotonicity as if p is not true then it is assumed p is false.

f

Abduction starts with statements we know to be true in a knowledge base and seeks to find the simplest clauses that make statement true.

Deduction starts with a knowledge base and seeks to deduce as many things as possible.

They are similar processes starting from opposite ends of the proof.