

CS-171, Intro to A.I. — Final Exam — Fall Quarter, 2012

NAME AND EMAIL ADDRESS: _____

YOUR ID: _____ ID TO RIGHT: _____ ROW: _____ NO. FROM RIGHT: _____

The exam will begin on the next page. Please, do not turn the page until told.

When you are told to begin the exam, please check first to make sure that you have all 10 pages, as numbered 1-10 in the bottom-left corner of each page.

The exam is closed-notes, closed-book. No calculators, cell phones, electronics.

**Please clear your desk entirely, except for pen, pencil, eraser, an optional blank piece of paper (for optional scratch pad use), and an optional water bottle.
Please turn off all cell phones now.**

This page summarizes the points available for each question so you can plan your time.

1. (10 pts total) Decision Tree Classifier Learning.
2. (5 pts total, -1 pt each wrong answer, but not negative) Search Properties.
3. (10 pts total) Naïve Bayes Classifier Learning.
4. (15 pts total, 5 pts each, -1 each error, but not negative) Bayesian Networks.
5. (10 points total, 2 pts each) Constraint Satisfaction Problems.
6. (10 pts total, -1 for each error, but not negative) Alpha-Beta Pruning.
7. (10 pts total, -2 for each error, but not negative) Conversion to CNF.
8. (10 pts total, -2 for each error, but not negative) Resolution Theorem Proving.
9. (10 pts total, 1 pt each) State-Space Search.
10. (10 pts total, 2 pts each) English to FOL Conversion.

The Exam is printed on both sides to save trees! Work both sides of each page!

See Section 18.3.

1. (10 pts total) **Decision Tree Classifier Learning.** You are a robot in a lumber yard, and must learn to discriminate Oak wood from Pine wood. You choose to learn a Decision Tree classifier. You are given the following examples:

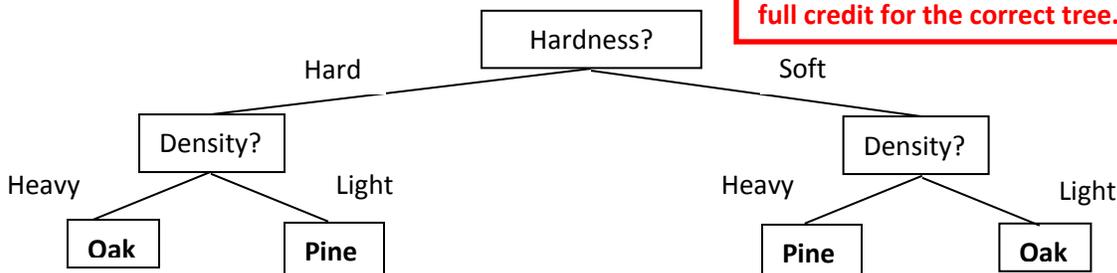
Example	Density	Grain	Hardness	Class
Example #1	Heavy	Small	Hard	Oak
Example #2	Heavy	Large	Hard	Oak
Example #3	Heavy	Small	Hard	Oak
Example #4	Light	Large	Soft	Oak
Example #5	Light	Large	Hard	Pine
Example #6	Heavy	Small	Soft	Pine
Example #7	Heavy	Large	Soft	Pine
Example #8	Heavy	Small	Soft	Pine

If root is Density:
 Heavy = OOOPPP, Light = OP
 If root is Grain:
 Small = OOPP, Large = OOPP
 If root is Hardness:
 Hard = OOOP, Soft = OPPP
 (O = Oak, P = Pine)

1a. (2 pts) Which attribute would information gain choose as the root of the tree?

Hardness

1b. (4 pts) Draw the decision tree that would be constructed by recursively applying information gain to select roots of sub-trees, as in the



Half credit for the correct root; half credit for wrong root but correct classification; full credit for the correct tree.

Full credit if your answers are right for the tree you drew, even if the tree itself is wrong.

Classify these new examples as Oak or Pine using your decision tree above.

1c. (2 pts) What class is [Density=Light, Grain=Small, Hardness=Hard]? Pine

1d. (2 pts) What class is [Density=Light, Grain=Small, Hardness=Soft]? Oak

2. (5 pts total, -1 pt each wrong answer, but not negative) **Search Properties.**

Fill in the values of the four evaluation criteria for each search strategy shown. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; C^* is the cost of the optimal solution; step costs are identical and equal to some positive ϵ ; and in Bidirectional search both directions use breadth-first search.

See Figure 3.21.

Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First	Yes	$O(b^d)$	$O(b^d)$	Yes
Uniform-Cost	Yes	$O(b^{(1+\text{floor}(C^*/\epsilon))})$ $O(b^{(d+1)})$ also OK	$O(b^{(1+\text{floor}(C^*/\epsilon))})$ $O(b^{(d+1)})$ also OK	Yes
Depth-First	No	$O(b^m)$	$O(bm)$	No
Iterative Deepening	Yes	$O(b^d)$	$O(bd)$	Yes
Bidirectional (if applicable)	Yes	$O(b^{(d/2)})$	$O(b^{(d/2)})$	Yes

See Sections 13. 5.2 and 20.2.2.

3. (10 pts total) Naïve Bayes Classifier Learning. You are a robot in a lumber yard, and must learn to discriminate Oak wood from Pine wood. You choose to learn a Naïve Bayes classifier. You are given the following (noisy) examples:

Example	Density	Grain	Hardness	Class
Example #1	Light	Small	Hard	Oak
Example #2	Heavy	Large	Hard	Oak
Example #3	Heavy	Small	Soft	Oak
Example #4	Heavy	Small	Soft	Oak
Example #5	Light	Large	Hard	Pine
Example #6	Light	Small	Soft	Pine
Example #7	Heavy	Large	Soft	Pine
Example #8	Light	Large	Hard	Pine

Recall that Baye's rule allows you to rewrite the conditional probability of the class given the attributes as the conditional probability of the attributes given the class. As usual, α is a normalizing constant that makes the probabilities sum to one.

$$P(\text{Class} \mid \text{Density, Grain, Hardness}) = \alpha P(\text{Density, Grain, Hardness} \mid \text{Class}) P(\text{Class})$$

3a. (2 pts) Now assume that the attributes (Density, Grain, and Hardness) are conditionally independent given the Class. Rewrite the expression above, using this assumption of conditional independence (i.e., rewrite it as a Naïve Bayes Classifier expression).

$$\alpha P(\text{Density, Grain, Hardness} \mid \text{Class}) P(\text{Class}) = \alpha P(\text{Density} \mid \text{Class}) P(\text{Grain} \mid \text{Class}) P(\text{Hardness} \mid \text{Class}) P(\text{Class})$$

3b. (4 pts total; -1 for each wrong answer, but not negative) Fill in numerical values for the following expressions. Leave your answers as common fractions (e.g., 1/4, 3/5).

$$P(\text{Oak}) = \underline{1/2}$$

$$P(\text{Pine}) = \underline{1/2}$$

$$P(\text{Density=Light} \mid \text{Class=Oak}) = \underline{1/4}$$

$$P(\text{Density=Light} \mid \text{Class=Pine}) = \underline{3/4}$$

$$P(\text{Density=Heavy} \mid \text{Class=Oak}) = \underline{3/4}$$

$$P(\text{Density=Heavy} \mid \text{Class=Pine}) = \underline{1/4}$$

$$P(\text{Grain=Small} \mid \text{Class=Oak}) = \underline{3/4}$$

$$P(\text{Grain=Small} \mid \text{Class=Pine}) = \underline{1/4}$$

$$P(\text{Grain=Large} \mid \text{Class=Oak}) = \underline{1/4} \quad P(\text{Grain=Large} \mid \text{Class=Pine}) = \underline{3/4}$$

$$P(\text{Hardness=Hard} \mid \text{Class=Oak}) = \underline{1/2} \quad P(\text{Hardness=Hard} \mid \text{Class=Pine}) = \underline{1/2}$$

$$P(\text{Hardness=Soft} \mid \text{Class=Oak}) = \underline{1/2} \quad P(\text{Hardness=Soft} \mid \text{Class=Pine}) = \underline{1/2}$$

3c. (2 pt each) Consider a new example (Density=Heavy ^ Grain=Small ^ Hardness=Hard). Write these class probabilities as the product of α and common fractions from above.

$$P(\text{Class=Oak} \mid \text{Density=Heavy} \wedge \text{Grain=Small} \wedge \text{Hardness=Hard}) = \underline{\alpha(3/4)(3/4)(1/2)(1/2)}$$

$$P(\text{Class=Pine} \mid \text{Density=Heavy} \wedge \text{Grain=Small} \wedge \text{Hardness=Hard}) = \underline{\alpha(1/4)(1/4)(1/2)(1/2)}$$

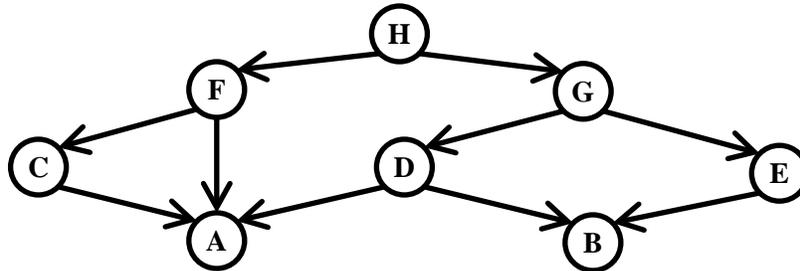
Here, $\alpha = 64/10$, and so converting likelihoods to probability yields:
 $P(\text{Class=Oak} \mid \text{Density=Heavy} \wedge \text{Grain=Small} \wedge \text{Hardness=Hard}) = 0.9$
 $P(\text{Class=Pine} \mid \text{Density=Heavy} \wedge \text{Grain=Small} \wedge \text{Hardness=Hard}) = 0.1$
 You do not need to do this, nor to provide a numeric value for α .

See Section 14.2.

4. (15 pts total, 5 pts each, -1 each error, but not negative) Bayesian Networks.

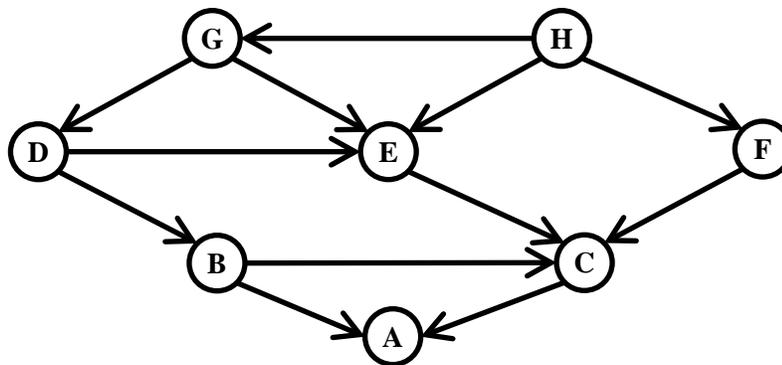
4a. (5 pts) Draw the Bayesian Network that corresponds to this conditional probability:

$$P(A | C,D,F) \cdot P(B | D,E) \cdot P(C | F) \cdot P(D | G) \cdot P(E | G) \cdot P(F | H) \cdot P(G | H) \cdot P(H)$$

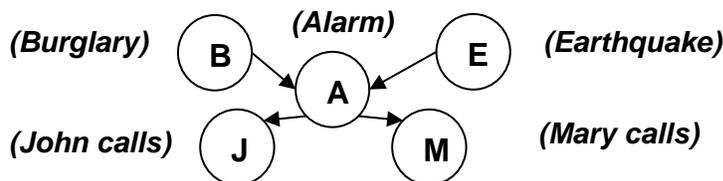


4b. (5 pts) Write down the factored conditional probability expression that corresponds to the graphical Bayesian Network shown.

$$P(A | B,C) P(B | D) P(C | B,E,F) P(D | G) P(E | D,G,H) P(F | H) P(G | H) P(H)$$



4.c. (5 pts) Shown below is the Bayesian network corresponding to the Burglar Alarm problem, $P(J | A) P(M | A) P(A | B, E) P(B) P(E)$.



P(E)
.002

A	P(M)
t	.70
f	.01

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

P(B)
.001

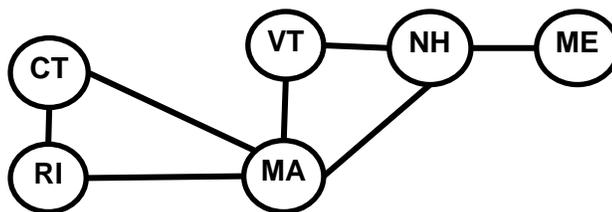
A	P(J)
t	.90
f	.05

The probability tables show the probability that variable is True, e.g., $P(M)$ means $P(M=t)$. Write down an expression that will evaluate to $P(j=t \wedge m=f \wedge a=f \wedge b=f \wedge e=t)$. Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols. You do not need to carry out the multiplication to produce a single number (probability). **SHOW YOUR WORK.**

$$\begin{aligned} &P(j=t \wedge m=f \wedge a=f \wedge b=f \wedge e=t) \\ &= P(j=t | a=f) * P(m=f | a=f) * P(a=f | b=f \wedge e=t) * P(b=f) * P(e=t) \\ &= .05 * .99 * .71 * .999 * .002 \end{aligned}$$

5. (10 points total, 2 pts each) Constraint Satisfaction Problems.

See Chapter 6.



You are a map-coloring robot assigned to color this New England USA map. Adjacent regions must be colored a different color (R=Red, B=Blue, G=Green). The constraint graph is shown.

5a. (2pts total, -1 each wrong answer, but not negative) FORWARD CHECKING.

Cross out all values that would be eliminated by Forward Checking, after variable MA has just been assigned value R as shown:

CT	RI	MA	VT	NH	ME
X G B	X G B	R	X G B	X R G B	R G B

5b. (2pts total, -1 each wrong answer, but not negative) ARC CONSISTENCY.

CT and RI have been assigned values, but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

CT	RI	MA	VT	NH	ME
R	G	XX B	R G X	R G X	R G B

5c. (2pts total, -1 each wrong answer, but not negative) MINIMUM-REMAINING-VALUES HEURISTIC.

Consider the assignment below. RI is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Minimum-Remaining-Values (MRV) Heuristic: CT, MA.

CT	RI	MA	VT	NH	ME
R B	G	R B	R G B	R G B	R G B

5d. (2pts total, -1 each wrong answer, but not negative) DEGREE HEURISTIC.

Consider the assignment below. (It is the same assignment as in problem 5c above.) RI is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Degree Heuristic: MA, NH.

CT	RI	MA	VT	NH	ME
R B	G	R B	R B	R G B	R G B

5e. (2pts total) MIN-CONFLICTS HEURISTIC.

Consider the complete but inconsistent assignment below. MA has just been selected to be assigned a new value during local search for a complete and consistent assignment. What new value would be chosen below for MA by the Min-Conflicts Heuristic? R.

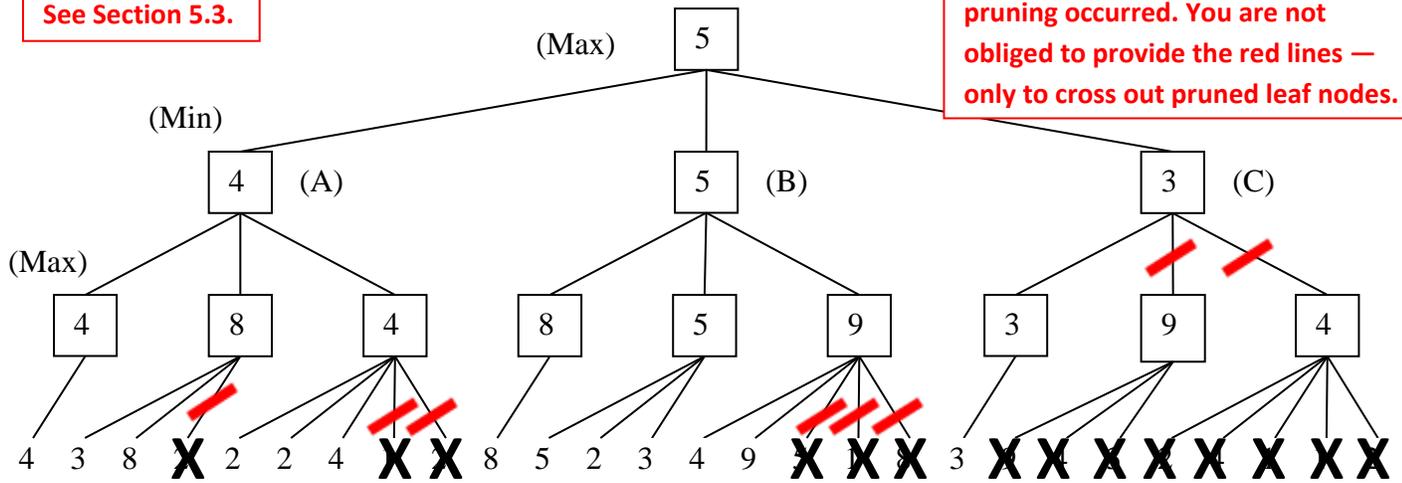
CT	RI	MA	VT	NH	ME
B	G	?	G	G	B

6. (10 pts total, -1 for each error, but not negative) Alpha-Beta Pruning. In the game tree below it is Max's turn to move. At each leaf node is the estimated score of that resulting position as returned by the heuristic static evaluator.

- (1) Perform Mini-Max search and label each branch node with its value.
- (2) Cross out each leaf node that would be pruned by alpha-beta pruning.
- (3) What is Max's best move (A, B, or C)? B

See Section 5.3.

Red lines indicate where in the tree pruning occurred. You are not obliged to provide the red lines — only to cross out pruned leaf nodes.



7. (10 pts total, -2 for each error, but not negative) Conversion to CNF. Convert this Propositional Logic wff (well-formed formula) to Conjunctive Normal Form and simplify. Show your work (correct result, 0 pts; correct work, 10 pts).

$$[\neg(Q \Rightarrow P)] \Leftrightarrow P$$

See Section 7.5.2.

/* convert $(A \Leftrightarrow B)$ into $(A \Rightarrow B) \wedge (B \Rightarrow A)$ */
 $\{[\neg(Q \Rightarrow P)] \Rightarrow P\} \wedge \{P \Rightarrow [\neg(Q \Rightarrow P)]\}$
 /* convert $(A \Rightarrow B)$ into $(\neg A \vee B)$ */
 $\{\neg[\neg(\neg Q \vee P)] \vee P\} \wedge \{\neg P \vee [\neg(\neg Q \vee P)]\}$
 /* apply DeMorgan's Laws, cancel double negations */
 $\{[(\neg Q \vee P)] \vee P\} \wedge \{\neg P \vee [(Q \wedge \neg P)]\}$
 /* distribute \vee over \wedge , simplify */
 $(\neg Q \vee P) \wedge (\neg P \vee Q) \wedge \neg P$
 /* distribute \wedge over \vee , simplify */
 $[(\neg Q \wedge \neg P) \vee (P \wedge \neg P) \vee (\neg Q \wedge Q) \vee (P \wedge Q)] \wedge \neg P$
 /* simplify */
 $[(\neg Q \wedge \neg P) \vee (P \wedge Q)] \wedge \neg P$
 /* distribute \wedge over \vee */
 $[(\neg Q \wedge \neg P \wedge \neg P) \vee (P \wedge Q \wedge \neg P)]$
 /* simplify */
 $(\neg Q \wedge \neg P)$

Some students did not simplify the final expression all the way, and so lost a point or two at the end. One popular stopping point was:
 $(\neg Q \vee P) \wedge (Q \vee \neg P) \wedge (\neg P)$
 However, a little algebra will convince you that further simplification is possible.

8. (10 pts total, -2 for each error, but not negative) Resolution Theorem Proving. You are a robot in a logic-based question answering system, and must decide whether or not an input goal sentence is entailed by your Knowledge Base (KB). Your current KB in CNF is:

- S1:** (P Q)
- S2:** (¬P Q)
- S3:** (P ¬Q)
- S4:** (¬P R)

See Section 7.5.2.

Your input goal sentence is: (P ∧ Q ∧ R).

8a. (2 pts) Write the negated goal sentence in CNF.

S5: _____ (¬P ¬Q ¬R) _____

It is OK to insert the disjunction symbol here.

8b. (8 pts total, -2 for each error, but not negative) Use resolution to prove that the goal sentence is entailed by KB, or else explain why no such proof is possible. For each step of the proof, fill in S_i and S_j with the sentence numbers of previous CNF sentences that resolve to produce the CNF result that you write in the resolvent blank. The resolvent is the result of resolving the two sentences S_i and S_j . Use as many steps as necessary, ending by producing the empty clause; or else explain why no such proof is possible.

Resolve S_i S1 with S_j S2 to produce resolvent **S6:** (Q)

Resolve S_i S1 with S_j S3 to produce resolvent **S7:** (P)

Resolve S_i S4 with S_j S7 to produce resolvent **S8:** (R)

Resolve S_i S8 with S_j S5 to produce resolvent **S9:** (¬P ¬Q)

Resolve S_i S7 with S_j S9 to produce resolvent **S10:** (¬Q)

Resolve S_i S6 with S_j S10 to produce resolvent **S10:** ()

Add additional lines below if needed; or, if no such resolution proof is possible, use the space below to explain why not:

Other proofs are OK if they are correct.

It is OK if you wrote the Boolean expression instead of S1, S2, etc. It is also OK if you put disjunction symbols inside the CNF clauses.

9. (10 pts total, 1 pt each) **State-Space Search.** Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as $h=x$). The successors of each node are indicated by the arrows out of that node.

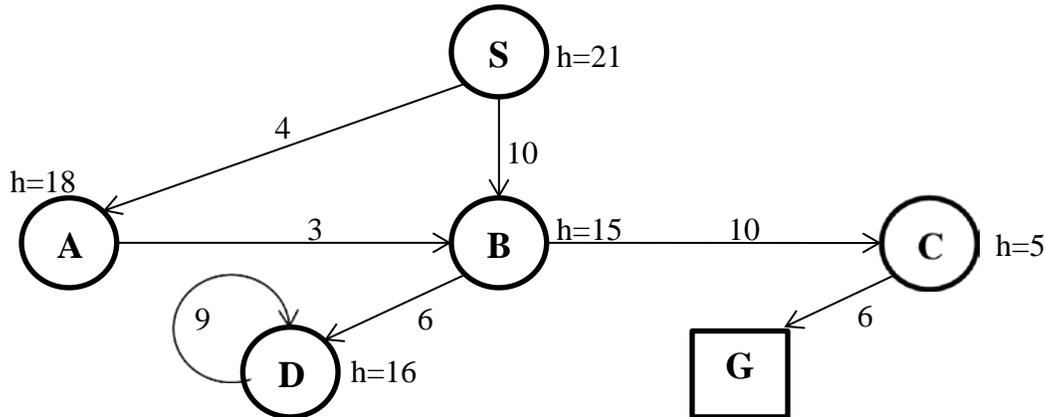
(Note: D is a successor of itself). As

The start node is S and the goal

(1) the order in which nodes are expanded

Write "None" for the path if the goal was not found.

Please see the lecture slides for Uninformed Search, topic "When to do Goal-Test? When generated? When popped?" for clarification about exactly what to do in practical cases.



9.a. DEPTH-FIRST SEARCH:

9.a. See Section 3.4.3 and Fig. 3.17. Path on: S A B D D D D ...

DFS can get caught in a loop in Tree Search (= do not remember visited nodes).

9.a.(2) Path to goal found: None

9.b. BREADTH-FIRST SEARCH:

9.b. See Section 3.4.1 and Fig. 3.11. Path on: S A B B D C G

BFS does the Goal-test before the child is pushed onto the queue. The goal is found when C is expanded.

9.b.(2) Path to goal found: S B C G

9.c. ITERATIVE DEEPENING SEARCH:

9.c. See Sections 3.4.4-5 and Figs. 3.18-19. Path on: S S A B S A B B D C G

IDS does the Goal-test before the child is pushed onto the queue. The goal is found when C is expanded.

9.c.(2) Path to goal found: S B C G

9.d. UNIFORM COST SEARCH:

9.d. See Section 3.4.2 and Fig. 3.14. Path on: S A B B D D C C D G

UCS does goaltest when node is popped off queue.

9.d.(2) Path to goal found: S A B C G

9.e. GREEDY BEST FIRST SEARCH:

9.e. See Section 3.5.1 and Fig. 3.23. Path on: S B C G

GBFS has the same behavior whether the goaltest is done before node is pushed or after node is popped, because $h=0$ for a goal node, so goal nodes always sort to the front of the queue anyway.

9.e.(2) Path to goal found: S B C G

9.f. A* SEARCH:

9.f. See Section 3.5.2 and Figs. 3.24-25. Path on: S A B C G

A* does goaltest when node is popped off queue.

9.f.(2) Path to goal found: S A B C G

10. (10 pts total, 2 pts each) English to FOL Conversion. For each English sentence below, write the FOL sentence that best expresses its intended meaning. Use $\text{Dog}(x)$ for “x is a dog,” $\text{Bone}(x)$ for “x is bone,” and $\text{Likes}(x, y)$ for “x likes y.”

The first one is done for you as an example.

See Section 8.2.6

10a. (2 pts) “Every dog likes every bone.”

$$\forall x \forall y [\text{Dog}(x) \wedge \text{Bone}(y)] \Rightarrow \text{Likes}(x, y)$$

10b. (2 pts) “Some dog likes some bone.”

$$\exists x \exists y \text{Dog}(x) \wedge \text{Bone}(y) \wedge \text{Likes}(x, y)$$

10c. (2 pts) “For every dog, there is a bone that the dog likes.”

$$\forall x \exists y \text{Dog}(x) \Rightarrow [\text{Bone}(y) \wedge \text{Likes}(x, y)]$$

10d. (2 pts) “For every bone, there is a dog who likes that bone.”

$$\forall y \exists x \text{Bone}(y) \Rightarrow [\text{Dog}(x) \wedge \text{Likes}(x, y)]$$

10e. (2 pts) “There is a bone that every dog likes.”

$$\exists y \forall x \text{Bone}(y) \wedge [\text{Dog}(x) \Rightarrow \text{Likes}(x, y)]$$

10f. (2 pts) “There is a dog who likes every bone.”

$$\exists x \forall y \text{Dog}(x) \wedge [\text{Bone}(y) \Rightarrow \text{Likes}(x, y)]$$