**CSci 384: Artificial Intelligence Spring, 2020**

Date: February 20th (Thr.), 2020

Due: by the end of day, March 5th (Thr.), 2020 -- **No Extension of Deadline**

**Midterm**

Name: Derek Trom **Total: 250 + 35 (optional) points**

Please answer all of the questions at your best.

You have to show the sufficient steps and explanation for your answers. -- Sloppy answers will NOT get a full point.

Write your answers ***to this file*** and upload it to ‘Submission’ section at eZ-LMS: NO .pdf file is accepted.

Hours taken to complete the exam: \_9\_\_ Hours \_30\_\_ Minutes.

Mark the difficulty of the Exam:

Very Easy: \_\_\_\_\_ Easy: \_\_\_\_\_ Moderate: \_\_\_\_\_ Difficult: \_X\_\_ Very Difficult: \_\_\_\_\_

Comment:

I feel like I have learned more about these subjects doing the take home exam rather than an in class exam.

**Q1. [20] Environment**

Give i) ***PEAS description*** of the task environment and ii) characterize it in terms of the ***properties*** of environment, e.g.) observable, deterministic, static, sequential, discrete, single-agent. Explain your answers.

1. [10] Playing a tennis against a wall.

Performance Measure = Hit speed and accuracy which will get better with time

Environment = Wall inside or outside which is where it will be played.

Actuator = Tennis racquet, tennis ball, arm

Sensor = Ball locator, camera, racquet sensor to sense if the player is accurately performing

It is observable because you can fully see action of the player hitting the ball and measure it, deterministic because only the players actions and the wall are taken into account, sequential and episodic because the agent can only respond according to the percept at each swing, dynamic depending on weather or swing, continuous because the agent can go as long as it is willing to go, and a single agent.

1. [10] Shopping for used AI books on Amazon.

Performance Measure = Price, condition, shipping time

Environment = Internet, sellers, shipping companies

Actuator = Search form, URL to certain books, user display

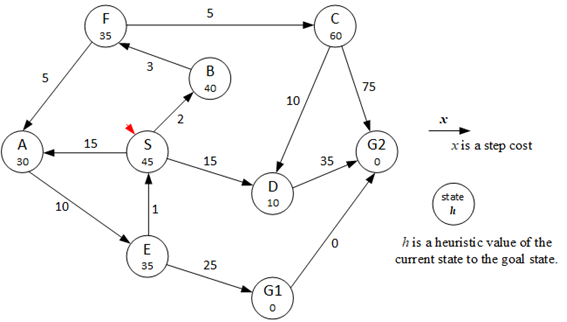
Sensors = The displayed results on the webpage.

It is partially observable because the agent can choose the books based on what is found but cannot the actual search functions embedded, stochastic the agent does not know the next state, it is semi-dynamic because as time passes certain books may not be available so the agents score for what is available will. It is discrete because there are a finite number of states that can be available meaning the book selection. This is also multi-agent including other buyers, the sellers and the shippers.

**Q2. [80] Search Strategies**

Assume you have the following search graph, where S is the start node and G1 & G2 are the goal nodes. Arcs are labeled with the step cost of traversing them and the estimated cost to a goal is reported inside nodes (i.e. *h*-value).

For each of the search strategies listed below, (a) indicate which goal state is reached (if any), (b) list, in order, all the states expanded (i.e. in the explored list), and (c) give a solution as a sequence of states. (Recall that a (non- goal) state expands its successor states when it is removed from the frontier list: then, a goal test is applied and is added to the explored list if it’s not a goal state.) When all else is equal, nodes will be expanded in alphabetical order. S is a start state and *a goal test is applied when a node is removed from the frontier list* for any search algorithm.



1. [10] Breadth-First Search:

Goal state reached: \_G2\_\_\_\_

State expanded: S,B,A,D,F,E

Solution: S,D. G2, 50 cost

1. [10] Depth-First search:

Goal state reached: \_\_G2\_\_

State expanded: S,B,F,C

Solution: S, B, F, C, G2 Cost = 85

1. [10] Uniform-Cost Search:

Goal state reached: \_\_\_G1\_\_

State expanded: S, B, F, A, C, D, E

Solution: S=0, B= 2, F=5, A = 10, E= 20, G1 = 45 total

1. [10] Iterative Deepening Search:

Goal state reached: \_G2\_\_\_\_

State expanded: S, A, B, D, E, F

Solution: S, D, G2 Cost = 0+15, 35 = 50

1. [10] Hill Climbing Search (using function *h* only):

Goal state reached: \_G2\_\_\_\_

State expanded: S, A, B, D, G2

Solution: S, D, G2 Cost = 0 + 15 + 35 = 50

1. [10] A\* Search: a goal test is applied when it’s removed from the frontier list.

For the states expanded, show both the *states* and their *f values*.

Goal state reached: \_\_G2\_\_\_

State expanded: S(45), D(25), B(42), F(40), A(40)

Solution: S, D, G2 Total Cost = 50

1. [10] Imagine that the simulated annealing algorithm is applied. A node ‘F’ is the randomly chosen current node and a node ‘A’ is the candidate successor. Assume that the current temperature T equals 5.
2. What is the probability that node A will be accepted as the next state?

F(f) – f(A) = 5 > 0 so it will be decided as e -∆E/T

= e35-30/5 = 36.7%

1. Assume that a node C is chosen as the candidate successor.

What is the probability that C will be accepted as the next state?

F(F) – F(C) = -25 which is < 0 so it will be accepted 100% as the next state.

1. [10] The heuristic path algorithm is a best-first search in which the objective function is

*f(n) = (2−w)g(n) + w·h(n).* For what values of *w* is this algorithm guaranteed to be optimal? Explain your answer. You may assume that *h* is admissible.

In order for this to be optimal w ≤ 1 because f(n) = (2-w)[g(n)+ (w/2-w)h(n)] and f(n) = g(n)+(w/(2-w))h(n) so in order for it to be optimal w/(2-w) ≤ 1 and w≤1.

**Q3. [20] Admissible Heuristic**

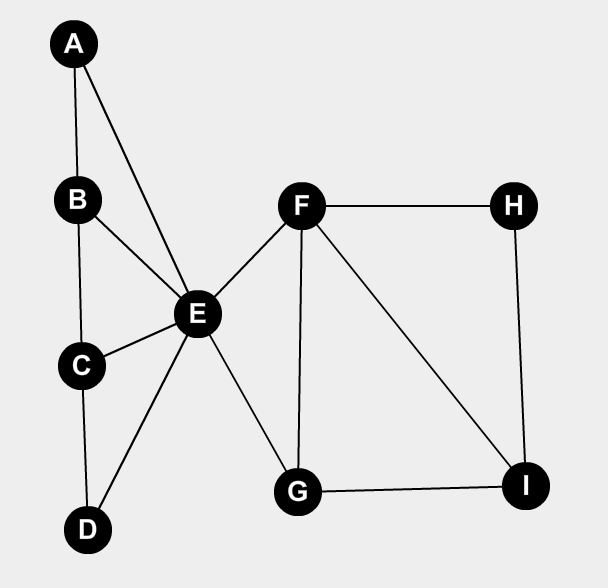
Assume you are given three admissible *h* functions for a given search problem; namely, *h1, h2*, and *h3*.

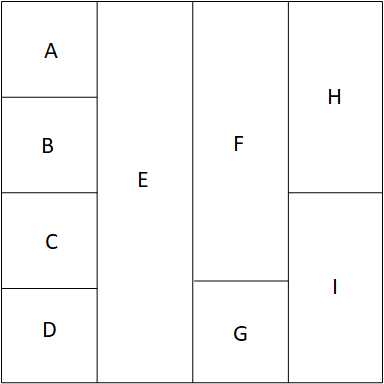
For each of the following combinations, explain whether or not the resulting *h* function is admissible.

* 1. Since h1(n) ≤ h\*(n) and h2(n) ≤ h\*(n) and h3(n)≤h\*(n) are all admissible is also admissible.
  2. Since h1(n) ≤ h\*(n) and h2(n) ≤ h\*(n) and h3(n)≤h\*(n) we can say that is admissible.

**Q4. [35] Constraint Satisfaction Problem**

We plan to plant the flowers on April in the given land in the figure. The bulbs of the following flowers are ready to plant: daffodil, lily, tulip. Since each sub-land, A – I, belongs to a different owner, any adjacent sub-land should be distinguished by planting a different bulb.

1. [5] Formulate the given problem.
   1. X = {A,B,C,D,E,F,G,H,I}
   2. D = {Daffodil, Lily, Tulip}
   3. C = {(A!=B)(A!=E)(B!=C)(B!=E)(C!=D)(C!=E)(D!=E)(E!=F)(E!=G)(F!=G)(F!= H)(F!=I)(G!=I)(H!=I)}
2. [5] Draw its constraint graph.
   1. 
3. [10] Starting from the current partial assignment {A=daffodil, B = lily}, find a solution manually using the following strategy: backtracking search with MRV heuristic, the degree heuristic, least-constraining value heuristic, forward checking and/or constraint propagation (AC-3). At each step, you have to clearly specify what strategy is applied to improve the efficiency.
   1. I would start by planting E with tulip using the MRV heuristic because it is touching the most boxes of all and A(Daffodil) and B(lily) are already planted which leaves only one option to plant.
   2. I would then plant C with daffodil using MRV because it only can be that as E is tulip and B = Lily.
   3. Using MRV the on D now because it only has one choice left I would plant D with lily.
   4. Using the degree heuristic, I would next set F to daffodil. Because it has the most constraints left of all.
   5. Then using MRV G would have to be set to lily next because it only has one choice
   6. Using MRV again set I to tulip it only has one choice
   7. Forward check to H and set to lily because that is the only choice left
   8. The final solution is {(A=Daffodil)(B=Lily)(C=Daffodil)(D=Lily)(E=Tulip)(F=Daffodil)(G=Lily)(H=Lily)(I=Tulip)}

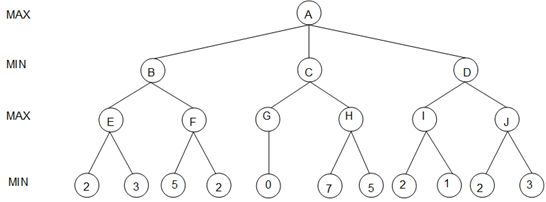


1. [5] If a constraint graph of problem can be reduced to a (nearly) tree-structured CSP, the problem may be solvable in linear time O(*n*) where *n* is the # of nodes. By conditioning method, decide a cycle cutset of the node(s) that need(s) to be instantiated.
   1. The cycle cutset first will be {A,B,C,D,E,F,G} where the decomposed will be {(A,E,B), (B,E,C), (C,E,D), (D,E,F), (F,E,G)}
   2. The second will be {FGHI} decomposed will be (F,G,I) and (FIH)
2. [10 ] Using your cycle cutset in (4), solve it in the nearly tree-structured CSP (and/or) in the decomposed subproblems (if it’s a case). Show your solution step by step with the changes of domains.
   1. For the cutset of {A,B,C,D,E,F,G}
      1. {(A=Daffodil, E= Tulip, B= Lily), (B=Lily, E=Tulip,C), (C,E=Tulip,D), (D,E=Tulip,F), (F,E=Tulip,G)}
      2. {(A=Daffodil, E= Tulip, B= Lily), (B=Lily, E=Tulip, C= Daffodil), (C=Lily, E=Tulip, D), (D=Daffodil, E=Tulip ,F), (F,E=Tulip,G)}
      3. {(A=Daffodil, E= Tulip, B= Lily), (B=Lily, E=Tulip, C= Daffodil), (C= Daffodil,E=Tulip,D=Lily), (D=Lilyl,E=Tulip,F), (F,E=Tulip,G)}
      4. {(A=Daffodil, E= Tulip, B= Lily), (B=Lily, E=Tulip, C= Daffodil), (C= Daffodil,E=Tulip,D=Lily), (D=Lilyl,E=Tulip,F =Daffodil), (F=Daffodil,E=Tulip,G)}
      5. {(A=Daffodil, E= Tulip, B= Lily), (B=Lily, E=Tulip, C= Daffodil), (C= Daffodil,E=Tulip,D=Lily), (D=Lilyl,E=Tulip,F =Daffodil), (F=Daffodil,E=Tulip,G= Lily)}
   2. Then choosing (F,G,H,I) as a cutset where F = Daffodil and G = Lily and decomposed are (F,G,I) and (FIH)
      1. {(F= Daffodil, G= Lily, I),(F = Daffodil, I, H)}
      2. {(F= Daffodil, G= Lily, I = Tulip), (F= Daffodil, I = Tulip, H)}
      3. {( F= Daffodil, G= Lily, I = Tulip), (F= Daffodil, I = Tulip, H = Lily )}
   3. This leaves the assignments {(A=Daffodil)(B=Lily)(C=Daffodil)(D=Lily)(E=Tulip)(F=Daffodil)(G=Lily)(H=Lily)(I=Tulip)}

**Q5. [35] The MiniMax algorithm and** **α-β** **Pruning**

Suppose that you’re a MIN player at the root A.

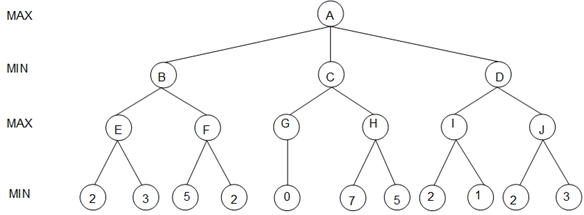
1. [10] (a) Decide in the ***minimax values*** at each node (A – J) in the Min-Max tree, and (b) ***mark the winning action*** for MIN player at the root A with a red line.
   1. A=3, B=3, C= 0, D=2, E= 3, F=5, G=0, H=7, I=2, J=3



1. [15] Given the same Min-Max tree below,
2. Show ***the changes of α or β values*** at each node,
   1. A = 3
   2. B = 3
   3. C= 0, C≤0
   4. D ≤ 2
   5. E = 2, E = 3
   6. F ≤ 5
   7. G = 0
   8. H =
   9. I = 2
   10. J =

(b) mark the ***branches that are pruned*** from the search by *α-β pruning*, and

(c) show the ***path*** chosen by MIN at the root.



1. [10] The effectiveness of *α-β pruning* search depends on the order of nodes in the Min-Max search tree. For the tree in (2), decides the best order and the worst order of nodes (B, C, D), respectively, and explain/justify your orders.
   1. The worst order by far is going to be (C, D, B) only 3 branches are pruned during the search.
   2. The can either be (B, C, D) or (B, D, C) 7 branches will be pruned making the search quicker than the other orderings.

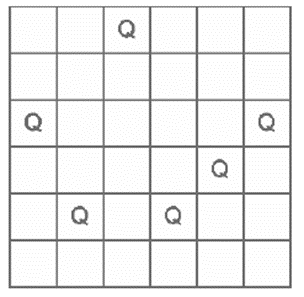
**Q6. [20] Simulated Annealing**

The *n*-queens problem requires you to place *n* queens on an *n* × *n* chessboard such that no queen attacks another queen. (A queen attacks any piece in the same row, column or diagonal.).

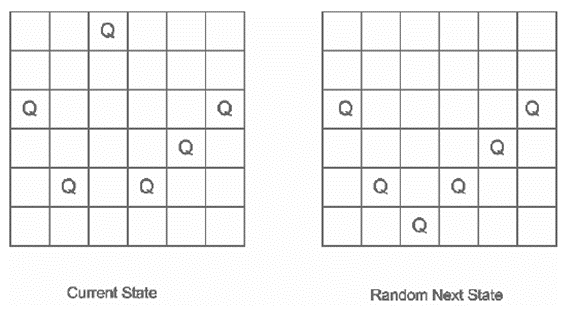
* We define the states to be any configuration where the *n* queens are on the board, one per column.
* The move set includes all possible states generated by moving a single queen to another square in the same column. The function to obtain these states is called the successor function.
* The evaluation function ***f*(*state*)** is the ***number of non-attacking pairs of queens*** in this state.

1. [5] How many possible states are there?
   1. 46656 possible states because there are N number of queens that can take N number of spots within a column so 66 = 46656.
2. [5] For each state, how many successor states are there in the move set?
   1. N\*(N-1) = 6\*(6-1) = 30

Suppose that a state is represented as a list of rows in which each queen is located from left. For instance, a state below is (3, 5, 1, 5, 4, 3).



1. [5] What value will the evaluation function *f* (•) return for the above current state of (2) in the figure? *f* (•) = 9 in the above state.
2. [5] If you use Simulated Annealing (currently T=3), and the current state and the random next state are shown below, will you accept this random next state immediately? or accept it with some probability? If it is the latter case, what is the probability? (f(next)=6)-(f(current)=9) = -3 so the move will be accepted e -3/3 = .3678 =36.78%



1. [10] Suppose that you use a Genetic Algorithm. The current generation includes four states, S1 through S4. The evaluation values for each of the four states may be computed. Calculate the probability that each of them would be chosen in the selection step.
   1. S1 = (4, 6, 2, 6, 4, 4) Pr(S1) = \_\_25.8%\_\_\_
      1. F(s1)= 8
      2. 8/(8+6+8+9) = 8/31 = .258
   2. S2 = (3, 5, 6, 5, 4, 3) Pr(S2) = \_\_19.4%\_\_\_
      1. F(S2) = 6
      2. 6/(8+6+8+9) = 6/31 = .194
   3. S3 = (4, 6, 2, 4, 6, 2) Pr(S3) = \_\_25.8%\_\_\_
      1. F(S3)=8
      2. 8/(8+6+8+9)= 8/31 = .258
   4. S4 = (6, 5, 1, 3, 6, 6) Pr(S4) = \_\_29.0%\_\_\_
      1. F(s4) = 9
      2. 9/(8+6+8+9)= 9/31 = .290

**Q7. [10] 3SAT and Hill-Climbing.**

SAT is the abbreviation for the satisfiability problem.

3SAT is the problem of finding a satisfying truth assignment for a sentence in a 3-CNF format, which is

defined as follows:

* A *literal* is a proposition symbol or its negation (e.g. *P* or *¬ P*).
* A *clause* is a disjunction of literals; a 3-clause is a disjunction of exactly 3 literals

(e.g. *P* ∨ *Q* ∨ *¬R*).

* A sentence in CNF or *conjunctive normal form* is a conjunction of clauses; a 3-CNF sentence is a conjunction of 3-clauses.

For example,

(*P* ∨ *Q* ∨ *¬ S*) ∧(*¬ P* ∨ *Q* ∨ *R*) ∧(*¬ P* ∨*¬ R* ∨ *¬ S*) ∧(*P* ∨*¬ S* ∨ *T*)

is a 3-CNF sentence with four clauses and five proposition symbols.

In formulating 3SAT problem, each state in the state space corresponds to an assignment of *True* or *False* to *every* propositional symbol. The successors of a state are defined as all those states which differ from the current state in the value of exactly one propositional symbol. The initial state is a random assignment. Assume that we use Hill-Climbing (without random restart) as the search strategy where the evaluation function used measures the number of satisfied clauses and may move to a neighbor with equal value.

1. [5] Given a 3-CNF sentence containing *n* distinct propositional symbols, how many successor states

should Hill-Climbing method evaluate from any given state? (Allow repeated states in the search tree.)

It should only evaluate n successor states from the current state because it only evaluates its neighbor states from the current states.

1. [5] Consider the 3-CNF sentence:

(*¬ P* ∨ *Q* ∨ *R*) ∧(*P* ∨ *¬ Q* ∨ *R*) ∧(*P* ∨ *Q* ∨ *¬ R*) ∧(*P* ∨ *Q* ∨ *R*)

1. [5] Define a state (or states) that satisfy the goal condition.
   1. (P= True, Q=True, R = True)
   2. (P= False, Q = True, R=True)
   3. (P= False, Q= False, R=True)
   4. (P= True, Q= False, R=True)
   5. (P= True, Q= True, R=False)
2. [5, optional] What does the Hill-Climbing algorithm do next from the state defined by

(*P* = *False, Q* = *False, R* = *False*)?

It will randomly select a propositional symbol P or Q or R to change to True and then generate successor states with that symbol being True then examine the current states ‘neighbor states’ and check for a goal state at each state.

**Q8. [20] Game Playing**

Consider the following nondeterministic game:

* When it is their turn to move, players must first choose which of two weighted coins,

A and B, to flip.

* Coin A comes up heads 10% of the time and tails the other 90%.

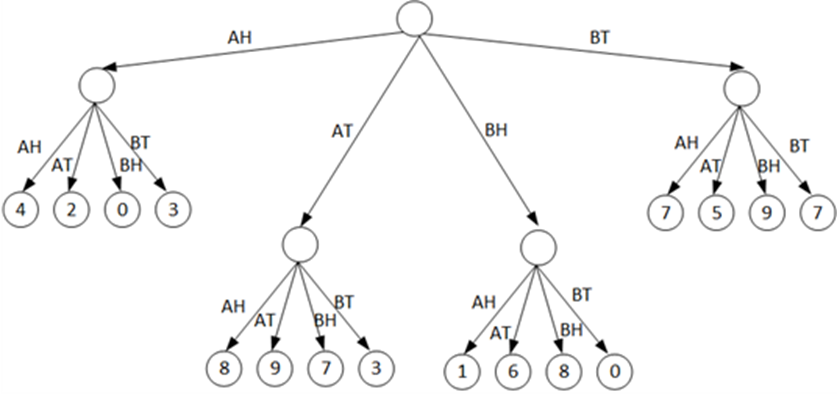
If heads, players must make move AH and if tails they must make move AT.

(To do this problem, you need not know exactly what each move means.)

* Coin B comes up heads 75% of the time and tails the other 25%.

If heads, the player must make move BH and if tails he or she (or it) must make move BT.

Assume it is the computers turn to play, and the game tree looks like the one below, where the values at the leaf nodes are the results of calls to the utility function F (higher scores are better for the computer).



1. [10] Explain what move the computer should make.

(Hint: think about expected-value calculations. Also, you might want to do parts it b and c first.)

AH = 1 \* 4 + .9 \* 2 + .75 \* 0 + .25 \* 3 = 2.95

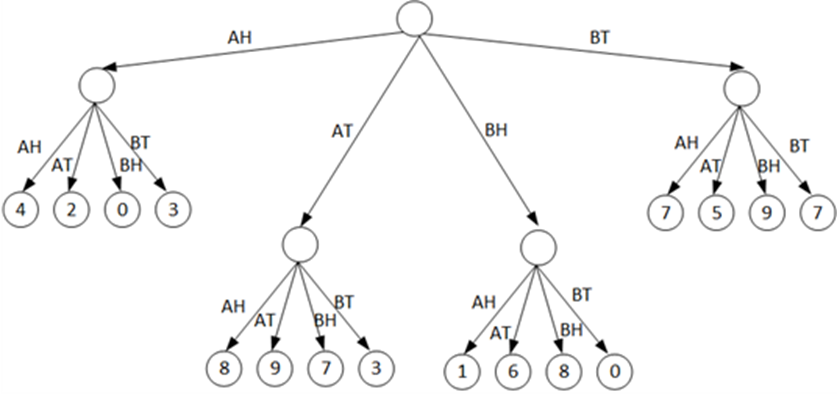
AT = .1 \* 8 + .9 \* 9 + .75 \* 7 + .25 \* 3 = 14.9

BH = .1 \* 1 + .9 \* 6 + .75 \* 8 + .25 \* 0 = 11.5

BT =.1 \* 7 + .9 \* 5 + .75 \* 9 + .25 \* 7 = 13.7

The computer should flip the A coin so it can try to move down the AT branch.

1. [10] Now assume that there is no randomness and the players simply can choose any of the four moves (AH, AT, BH, or BT). Apply the minimax algorithm to the tree below and explain which move the computer should make. As in part (a), assume it is the computers turn to play.
   1. The computer should select the BT because it would be the highest score of 5.



**Q9. [30, optional] Cryptarithmetic Problem**

For the given cryptarithmetic problem,

EUROPA + MARS + URANUS = SATURN

1. [10] Formulate the given problem.
   1. EUROPA

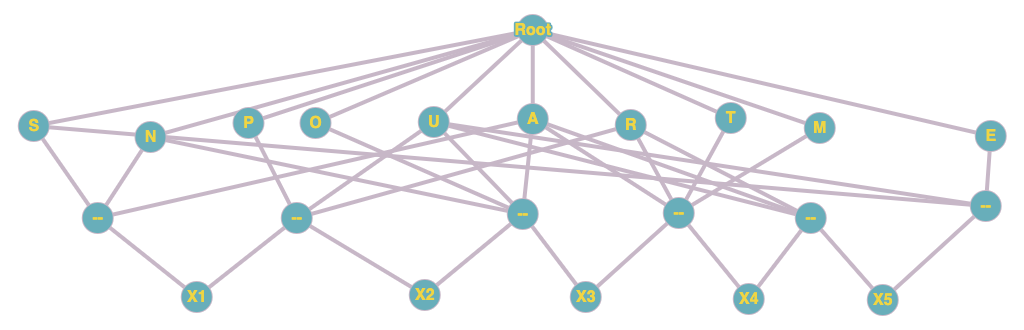
MARS

URANUS +

SATURN

* 1. Variables = {A, E, M, N, O, P, R, S, T, U, X1, X2, X3, X4, X5}
  2. Domain = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  3. Constraints = AllDiff(A,E,M,N,O,P,R,S,T,U)
     1. A + S + S = 10\*X1 + N
     2. P + R + U + X1 = 10\*X2 + R
     3. O + A + N + X2 = 10\*X3 + U
     4. R + M + A + X3 = 10\*X4 + T
     5. U + R + X4 = 10\*X5 + A
     6. E + U + X5 = S
     7. E! =0, M!=0, U!=0, S!=0

1. [5] Draw its constraint graph.



1. [15] Solve the problem, using backtracking search with constraint propagation (AC-3), MRV heuristic, the degree heuristic, and/or least-constraining-value heuristic, etc. Show each step of your solution by specifying the domains of variables with the remaining values and specifying which heuristic/algorithm is applied to improve the efficiency.

EUROPA + MARS + URANUS = SATURN

431865 + 2517 + 315937 = 750319

A = 5 , E = 4, M = 2, N = 9, O = 8, P = 6, R = 1, S = 7, T = 0, U = 3

Output of program below

