Introduction to Artificial Intelligence

1. Agent Architectures

- agent perceives environment, acts according to performance criteria
- reflexive agents: perceive → action rules → action
  (table lookup agent)
- agent with internal world model: actualize world model from
  what they perceive + what their actions do
- agents with explicit goals: choose ~~so~~ actions to achieve goals
- utility-based agents: choose actions based on utility of new state
- Learning agents: has critic + learning element to evaluate
  performance and find better actions

environment properties:
accessible/nonaccessible, deterministic/nondeterministic
episodic/nonepisodic, static/dynamic, discrete/continuous

2. Search

- used in goal-oriented agents to find path to goal
- problems: single-state, multiple-states, contingency, exploration
- choosing state space: abstraction, avoid non-intended solutions
- evaluating strategies: completeness, time complexity,
  space complexity, optimality
- Breadth-first search: complete, (evtl.) optimal, time + space $O(b^{d+1})$
- uniform cost search: same as BFS + optimal
- Depth-first search: not complete + optimal, time $O(b^n)$ space $O(b \cdot n)$
- Depth-limited search: DFS to max depth
- Iterative Deepening: optimal + complete, time $O(b^d)$ space $O(b \cdot d)$
- Bidirectional search: time + space $O(b^{d/2})$
  problems: operators need to be reversible, many goal states/badly defined

3. Informed Search

- evaluation function f: assigning each node real number
- h = heuristic of cost to goal state
- Best-First-search: choose node wit best f
- Greedy: choose node with best h (not optimal, not complete
- A^*: g = actual cost to node, $f = g + h$
  - heuristic admissible if $h(n) \leq h^*(n)$ (= real costs)
  - Path-Max equation: n parent of $n' \Rightarrow f(n') = \max(f(n), g(n') + h(n'))$
- Find heuristic as solution to simplified problem
- compute exact solution to subproblem
- IDA^*: combination of iterative deepening and A^*
- SMA^*: simplified memory bounded A^*: only save limited no. of nodes
- Hill Climbing: greedily choose best neighbour (finds local maximal
  - simulated annealing: take downward step with certain prob.

4. Games

- features: early pruning, good evaluation function
- minimax: complete game tree, utility function at terminal states
  choose min or max depending on state
  - can use cut-off test instead of evaluating whole tree
  - evaluation function: easy to compute, reflect chance of winning
  - hard to decide when to cut off (horizon problem)
- AlphaBeta: don't evaluate whole tree depending on known vals
  - ideal case $O(b^{m/2})$ on average $O(b^{3/4 n})$
- possible to introduce chance

5. Knowledge Representation

- Levels: Knowledge, Symbolic, Implementation
- language: declarative, precise $\Rightarrow$ First Order Logic
- Alphabet: Delimiter (), Operators: $\neg, \wedge, \ldots$, Variables: $x_1, x_2, \ldots$
  Predicate Symbols, function Symbols

- Grammar: Terms: variables + $f(t_1 .. t_n)$ atomic wffs: $P(t_1, .. t_n)$ $t_1 = t_n$

  formulas: $\neg\alpha$, $(\alpha \wedge \beta)$, $\exists x \alpha$ ....
- variables can be free: $P(x)$ or bound $\exists x P(x)$
- sentences: wffs without free variables
- substitution: $\alpha[x/t]$
- Interpretation: $I = \langle$ Domain, function $\phi \rangle$ $\phi(P) \subseteq D \times .. D$ $\phi(f) \in [D \times .. D \to D]$

  $\phi(P) = \{\}$ false $\phi(P) = \{\langle\rangle\}$ true
- Denotation: $I, \nu \| x \| = \nu(x)$ $I, \nu \| f(t_1 .. t_n) \| = H(d_1, .. d_n)$
- Satisfaction $I, \nu \models \alpha$ α is satisfied by I and ν
- Logical Consequence: $S \models \alpha$ iff $\forall I$ $I \models S \Rightarrow I \models \alpha$ or $\forall I$ $I \not\models S \cup \neg\{\alpha\}$
- Deductive Inference: given KB and query α check wether $KB \models \alpha$
  - correct / complete

6. Resolution
- Formulars are set of Clauses (= set of Literals (= atomic sentence or neg.))
  - formula conjunction of clause, clause disjunction of literals
- convert into cnf: 1. eliminate $\supset, \equiv$ 2. push $\neg$ inwards 3. distribute

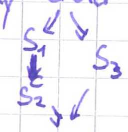
  $\vee$ over $\wedge$ 4. simplify
- inference rule: input clauses: $\{p\} \cup C_1$, $\{\neg p\} \cup C_2$ resolvent $C_1 \cup C_2$
- Derivation: if $S \to c$ then $S \models c$ (opposite does not hold)
- transform $KB \models \alpha$ to $S = KB \cup \{\neg\alpha\}$
  - if $S \to []$ unsatisfiable, if no mor resolution possible satisfiable
- Quantifiers: Interpretations are understood to be univresally quantified
- FOL $\to$ CNF extra steps: rename variables, eliminate $\exists$, move $\forall$ left
- answer predicate $\exists x P(x)$ to $\exists x [P(x) \wedge \neg A(x)]$ $\Rightarrow$ infer answer predicate
- skolemization: elimate $\exists$ by replacing variables with functions
  - not equivalence preserving but satisfiability remains
- resolution can be undecidable, but is complete

Most General Unifier: only unify when necessary

- Herbrand: transform FOL tho propositional sentences
  - universe: set of all possible terms
  - base: replace variables by terms of universe
- * - S satisfiable iff Herbrand base satisfiable
- Resolution can take exponential time
  - leave search control to user or use less expressive language

7. Planning

- Strips: actions have names, preconditions, effects
  - furthermore: initial state + goal state
- Plan: partially ordered steps, variable assignments $x=t$
  casual relations
  - complete plan: every precondition of every step is satisfied
  - consistent plan: $S_i \prec S_j$ then $S_j \not\prec S_i$ and $x=A$ then $x \neq B$
- algorithm: take plan step with unsatisfied preconditions →
  insert new plan step. Delay order as long as possible
- Protection of Causal relations



  - demotion $S_3 \neg c \to S_1 \to_c S_2$
  - promotion $S_1 \to_c S_2 \to S_3 \neg c$
- pop algorithm

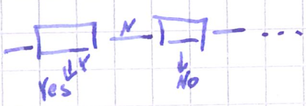
8. Uncertainty

- $P(A|B) = \frac{P(A \wedge B)}{P(B)}$ $\qquad P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$
- with additional evidence: $P(Y|X,E) = \frac{P(X|Y,E) \cdot P(Y|E)}{P(X|E)}$
- normalization $P(Y|X) = \alpha \cdot P(X|Y) \cdot P(Y)$ often suffizes
- $\mathbf{P}$ denote probability distribution
- Bayesian update. $P(C|T \wedge A) = P(C) \cdot \frac{P(T|C)}{P(T)} \cdot \frac{P(A|\cancel{T_{\wedge}}C)}{P(A|T)}$
  - given conditional independence
- only safe belief networks: acyclic graph with $\mathbf{P}(X|Parents(X))$
  - correct representation if $P(X_i|X_{i-1}, \dots X_1) = P(X_i|Parents(X_i))$

- E d-seperates X from Y, then X is independent of Y given E
  - d-separation if iff for every path from X to Y either
    1. $X - O \rightarrow \overset{E}{\boxed{Z}} \rightarrow O - Y$
    2. $X - O \leftarrow \overset{E}{\boxed{Z}} \rightarrow O - Y$
    3. $X - O \rightarrow \overset{\text{not } E}{Z} \leftarrow O - Y$
  - not every conditional independence is detected

9. Learning

- supervised: input and output avaible
- reinforced: feedback as punishment or reward
- unsupervised: no indication of correct output
- Decision trees: Input: Properties Output: Yes/No
  - every boolean function is representable
  - try to find most compact DT => choose best possible attribute
- training and test set seperate
- Information content $I(P(v_1) \dots P(v_n)) = \sum_{i=1}^{n} - P(v_i) \cdot \log_2 P(v_i)$
- Learning as elimination of hypotheses (search space too big)
  - remove hypothesis with false negatives or positives
- use only one hypothesis
  - use specialization and generalization
  - all previous examples need to be tested again
  - hard to find good heuristic
- PAC: how many examples needed so that h is within ε of f with prob. 1-δ
- Decision Lists: one test after another



  - each test has bounded literals

10. Neural Networks

- massive parallelism, robust, graceful degradation, inductive learning
- Input Function, activation function g, output a

  $a_i = g(in_i) = g\left(\sum_j w_{ji} \cdot a_j\right)$

  - g usually non linear: step function, sign, sigmoid
- topologies: feed-forward or recurrent

  recurrent → Hopfield nets, Boltzmann Machines
- FF nets: - 1 hidden layer → every continuous function
  - 2 " → every function
  - no hidden layer → perceptron (piece linearly separable functions)
- size difficult to find: iteratively reduce connections
- Perceptron learning: updater rule = $W_j := W_j + \alpha \cdot I_j \cdot Error$
- Back propagation:
  - output layer: $W_{j,i} = W_{j,i} + \alpha \cdot a_j \cdot \underbrace{Err_i \cdot g'(Input_i)}_{\Delta_i}$
  - hidden layer $W_{kj} = W_{kj} + \alpha \cdot I_k \cdot \Delta_j$

    $\Delta_j = g'(in_j) \cdot \sum_i W_{ji} \Delta_i$

11. Robotics

- geometric maps: cells $\langle x,y \rangle$ with probability $P(occ_{x,y})$
  - initialize with 0.5 then calculate as $hits/_{hits + misses}$
    - works for thin beams like lasers
  - for sonar use neural net with 4 sonar units + coordinates as input
- Pathplanning: iterate $V_{xy} = \min\{V_{x+\delta,y+\varepsilon} + \underset{\uparrow}{C}(\langle x+\delta, y+\varepsilon \rangle)\}$ for $\delta, \varepsilon \in \{-1,0,1\}$
- Probabilistic Localization: estimate $P(L_{xy\theta}|m) \hat{=} P(occ_{xy})$
  - initialize as $\frac{1-P(occ_{xy})}{\sum_{x'y'} 1-P(occ_{xy})}$
  - new position by integrating movement $P(L) := \alpha \cdot \sum_{L'} P(L') \cdot P(L|L',t)$
  - by sensor integration $P(L|s^T,\ldots,m) = P(L|s^{T-1},\ldots m) \cdot P(s^T|L,m)$

  $P(s|L,m) = \underset{\text{prob of one reflection in } R_i}{\underset{\shortparallel}{P(R_i|L,m)}} \cdot \prod_{j=1}^{i-1} 1 - P(R_j|L,m) \quad \Big| \quad P(R_i|L,m) = 1 - \prod_{L' \in R_i} (1 - \underset{\text{Prob that } L' \text{ reflects}}{\underset{\shortparallel}{P(r_{L'}|L,m)}})$

Problem Classification

L: move Left R: move right S: suck dust

1. $5 \xrightarrow{R} 6 \xrightarrow{S} 8$
2. $\{1,2,3,\dots 8\} \xrightarrow{L} \{1,3,5,7\} \xrightarrow{S} \{5,7\} \xrightarrow{R} \{6,8\} \xrightarrow{S} \{8\}$
3. suppose sometimes L does not work + no sensors

   $4 \xrightarrow{L} \{3,4\} \xrightarrow{S} \{7,4\} \xrightarrow{L} \{3,4,7\}$

   add dirt sensor

   $4 \xrightarrow{L} \{3,4\} \xrightarrow{DS} \cdot \xrightarrow{r} 3 \xrightarrow{S} 7$

   N

4. unknown world state + wallsensor

   $\{3,4,5\} \xrightarrow{L} \{3,4,5\} \xrightarrow{WS} \cdot \xrightarrow{r} \{3,5\} \xrightarrow{S} \{7,5\} \xrightarrow{R} \{6,8\} \xrightarrow{S} \{8\}$

   N

Theorem: A^* is optimal

Proof: suppose not. Then A^* chose a goal G' which is not optimal. Let G be the actual optimal goal. Let n be a leaf-node in the search tree, which lies on the optimal path to G. Let f^* be the optimal cost from the start node to G. Then $g(G') = f(G') > f^*$. Since n was not chosen for expansion $f(n) \geq f(G')$ and $f^* \geq f(n)$

$\Rightarrow f^* \geq f(G') = f^* \geq g(G')$ ↯

A^* is optimally efficient

For every search method M there exists a search space where M expands at least as many nodes as A^*

IDA^* is good if many nodes have the same f-cost o.w. bad

Properties SMA* (simplified memory-bounded)

- like A^* until memory is used up
- complete provided there is enough memory for shallowest solution
- optimal if enough memory for shallowest optimal solution

Hill climbing extensions

- random restart (use many random start nodes)
- Beamsearch: start with k start nodes compute all successors collect k best ones. iterate

$KB \overset{?}{\models} W_{1,3}$ show: for all I if $I \models KB$ then $I \models W_{1,3}$

Let $I \models KB$ then $I \models \neg S_{1,1}$ and $I \models$ "R_1"

then $I \models \neg W_{1,1}$, $I \models \neg W_{1,2}$

$I \models \neg S_{2,1}$ and $I \models$ "R_2" then $I \models \neg W_{2,2}$

$I \models S_{1,2}$ and $I \models$ "R_4" hence $\boxed{I \models W_{1,3}}$ or $I \models \cancel{W_{1,2}}$

or $I \models \cancel{W_{2,2}}$ or $I \models \cancel{W_{1,1}}$

example for exponential blowup in MGU computation

$L_1 = P(x_1, \dots, x_n)$

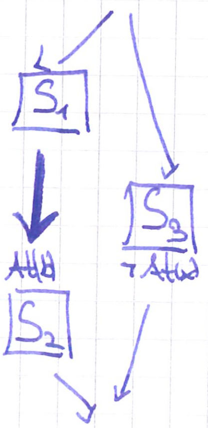
$L_2 = P(f(x_0,x_0), f(x_1,x_1), \dots, f(x_{n-1},x_{n-1}))$

$DS = \{x_1, f(x_0,x_0)\}$

$\Theta: x_1 / f(x_0,x_0)$

$\Theta: x_2 / f(f(x_0,x_0), f(x_0,x_0))$

$\vdots$



Threat? x variable
b constant

Ways of dealing with it:

1) choose $x := c$ with $c \neq b$
disadvantage: need to commit to c

2) extend language to include inequalities like $x \neq b$

3) ignore threat until the value of x is needed

SELECT-SUBGOAL never leads to backtracking
Why? Every ordering of subgoal selection will lead to
a solution, if one exists

STRIPS planning is PSPACE-complete

$$P(X=x) = \sum_y (P(X=x \wedge Y=y)) \quad \hat{=} \text{ Marginalization}$$
$$= \sum_y (P(X=x \mid Y=y) \cdot P(Y=y)) \quad \hat{=} \text{ Conditioning}$$

$$P(J \wedge M \wedge \neg B \wedge \neg E \wedge A) = P(J|A) \cdot P(M|A) \cdot P(A|\neg B, \neg E) \cdot P(\neg E) \cdot P(\neg B)$$

Suppose there are $n=20$ variables and each node has on average $k=5$ parents. The joint distribution has $2^n > 1000000$ entries. the graph has size $\sim n \cdot 2^k = 640$

$I(X,E,Y)$ means E d-seperates X from Y (i.e. X is independent of Y when E is given)
$I(\text{Gas}, \text{Ignition}, \text{Radion})$ = case 1 $I(\text{Gas}, \text{Battery}, \text{Radio})$ case 2
$I(\text{Gas}, -, \text{Radio})$ case 3
$\neg I(\text{Gas}, \text{Starts}, \text{Radio})$: $P(\neg\text{Gas} \mid \neg\text{Starts}, \text{Radio}) > (\neg\text{Gas} \mid \neg\text{Starts})$

$P(B|S)$: "John is 90% accurate and alarm is a good indication for B Hence $P(B|S)$ should be high"
But: Earthquakes twice as likely as Burglary
+ there is 1 B ind 1000 days but 50 false calls by J
$\Rightarrow$ $P(B|S) = 0{,}016$

$$P(b|j,m) = \alpha P(b,j,m)$$
$$= \alpha \sum_e \sum_a P(b,e,a,j,m) = \alpha \sum_e \sum_a P(b) \cdot P(e) \cdot P(a|b,e) \cdot P(j|a) \cdot P(m|a)$$
$$= \alpha P(b) \cdot \sum_e P(e) \cdot \sum_a P(a|b,e) \cdot P(j|a) \cdot P(m|a)$$

h is approximately correct if $Error(h) \leq \varepsilon \quad 0 < \varepsilon < 1$

$H_{Bad} = \{h_b \mid Error(h_b) > \varepsilon\}$

$P(h_b \text{ agrees with 1 example}) \leq (1-\varepsilon)$

$P(h_b \text{ agrees with } m \text{ examples}) \leq (1-\varepsilon)^m$

$P(H_{Bad} \text{ contains a hypothesis consistent with } m \text{ ex.}) \leq |H_{Bad}| \cdot (1-\varepsilon)^m$

$\leq |H| \cdot (1-\varepsilon)^m$

Want: $|H| \cdot (1-\varepsilon)^m \leq \delta$

$\Rightarrow m \geq 1/\varepsilon \, (\ln 1/\delta + \ln |H|) \quad (\text{use } 1-\varepsilon \leq e^{-\varepsilon})$

$|H| = 2^{2^n} \Rightarrow \ln(|H|) \approx 2^n$

Let k-$DL(n)$ be the language of decision lists over n attributes with $\leq k$ literals in a test

Want: $|k\text{-}DL(n)|$

Consider $Conj(n,k)$, the language of tests with $\leq k$ literals

First we use a DL on a set of tests

For each test T there are 3 cases

1) $\rightarrow \boxed{T} \rightarrow$ with $\downarrow$ Yes 2) $\rightarrow \boxed{T} \rightarrow$ with $\downarrow$ No 3) T does not occur in DL

Thus there are $3^{|Conj(n,k)|}$ sets of tests

Since the ordering matters: $|k\text{-}DL(n)| \leq 3^{|Conj(n,k)|} \cdot |Conj(n,k)|!$

$|Conj(n,k)| = \sum_{i=0}^{k} \binom{2n}{i} \approx O(n^k)$

Thus $|k\text{-}DL(n)| \approx 2^{O(n^k \cdot \log_2(n^k))}$

Then $m \geq 1/\varepsilon \, (\ln 1/\delta + O(n^k \cdot \log_2(n^k)))$

$\rightsquigarrow$ # examples needed polynomial in k

Input units have real values. How do you encode discrete inputs

E.g. Patrons attribute has 3 values: "none", "some", "full"

1) Local encoding (1 unit) 2) distributed encoding 3 units with values 0,1

none $\hat{=}$ 0.0	$I_1 \hat{=}$ none	0	
some $\hat{=}$ 0.5	$I_2 \hat{=}$ some	1	"some"
full $\hat{=}$ 1.0	$I_3 \hat{=}$ full	0	

Update rule

$$g\left(\sum W_j I_j\right) \Rightarrow g\left(\sum (W_j + \alpha \cdot I_j \cdot E) \cdot I_j\right)$$

Suppose $E > 0$ (O is too small)

if $I_j = 0$ no change

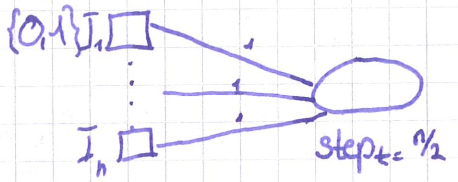
$I_j < 0$ then W_j decreases (negative influence of I_j decreases)

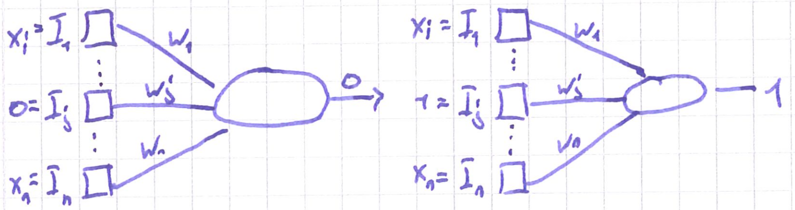
$I_j > 0$ then W_j increases (positive influence of I_j increases)

xor not learnable

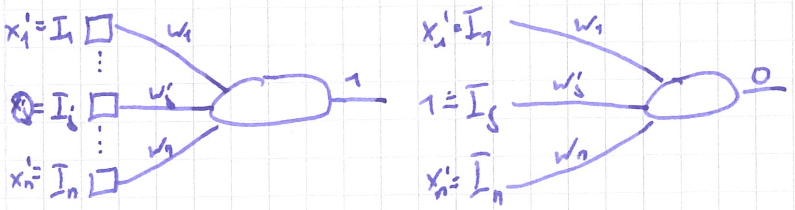
$$g(0,0) = 0 \quad \left.\begin{array}{l} g(1,0) = 1 \Rightarrow W_1 \geq t \\ g(0,1) = 1 \Rightarrow W_2 \geq t \end{array}\right\} g(1,1) = 1 \quad ↯$$

majority function

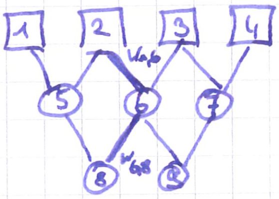




$\Rightarrow W_j > 0$



is impossible



$$E_{rr} = \frac{1}{2} \sum_i (T_i - O_i)^2$$

1. Output Layer update

$$W_{j,i} := W_{j,i} + \alpha \cdot a_j \cdot \underbrace{\overset{(T_i - O_i)}{Err_i} \cdot g'(\mathrm{Input}(O_i))}_{\Delta_i}$$

Note: if $g'(\mathrm{Input}(O_i))$ is near 0 then change $W_{j,i}$ very little

2. hidden Layer update

$$W_{k,j} := W_{k,j} + \alpha \cdot I_k \cdot \Delta_j$$

⇓ Back propagation $\Delta_j := g'(\mathrm{Input}(a_j)) \cdot \sum_i W_{j,i} \cdot \Delta_i$

estimate error portion which hidden unit a_j is responsible for

Let w be the vector of all weights in the NN

$$E(w) = \frac{1}{2} \sum_i (T_i - O_i)^2 = \frac{1}{2} \sum_i \left(T_i - g\left(\sum w_{j,i} \cdot a_j\right)\right)^2$$

$$= \frac{1}{2} \sum_i \left(T_i - g\left(\sum w_{j,i} \cdot g\left(\sum_k w_{k,j} \cdot I_k\right)\right)\right)^2$$

$$\frac{\partial E}{\partial w_{j,i}} = -a_j (T_i - O_i) \cdot g'\left(\sum w_{j,i} \cdot a_j\right) = -a_j (T_i - O_i) \cdot g'(\mathrm{Input}(O_i))$$

$$\frac{\partial E}{\partial w_{k,j}} = -I_k \cdot \Delta_j$$

Deep Learning

Each Layer focuses on differend features / can be pre-trained

In the end only refinement

sensor reading:

$$P(occ_{xy} \mid S_T, S_{T-1}, \dots) = \alpha \cdot P(S_T \mid occ_{xy}, S_{T-1} \dots S_1) \cdot P(occ_{xy} \mid S_{T-1} \dots S_1)$$

(conditional independence: $P(S_T \mid occ_{xy}, S_{T-1} \dots S_1) = P(S_T \mid occ_{xy})$

$$\Rightarrow = \alpha \left(P(S_T \mid occ_{xy})\right) \cdot P(S_{T-1} \mid occ_{xy}) \cdots P(S_1 \mid occ_{xy}) \cdot \underbrace{P(occ_{xy})}_{= 1/2}$$

errorrate of the sensor

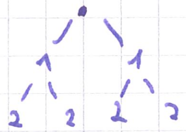
E 1.4

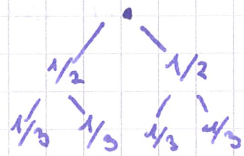
[RD | D] → [R | D] → [| RD] → [| R]

[D | RD] → [D | R] → [RD |] → [R |]

E 2.1

a) $h(n) = g(n)$  (Dijkstra) uniform cost search

b) $h(n) = d(n)$  breadth-first search

c) $h(n) = \frac{1}{1+d(n)}$  depth-first search

E 2.2 $h(n) \leq c(n,a,n') + h(n')$ To show: $h(n_0) \leq h^*(n_0)$

Proof:

1) Let $n_0 \xrightarrow{a_1} n_1 \xrightarrow{a_2} \dots \xrightarrow{a_L} n_L$ be a cheapest path from n_0 to a goal.

i) $h(n_L) = 0 \quad h^*(n_L) = 0 \Rightarrow h(n_L) \leq h^*(n_L)$

ii) $h^*(n_0) = \sum_{i=1}^{L} c(n_{i-1}, a_i, n_i) \geq \sum_{i=1}^{L} (h(n_{i-1}) - h(n_i)) = h(n_0) - h(n_L) = h(n_0)$

$c(n_{i-1}, a, n_i) \geq h(n_{i-1}) - h(n_i)$ by assumption

2) No path from n_0 to a goal exists

$\Rightarrow h^*(n_0) = \min\{c \mid c = \text{cost of a path from } n_0 \text{ to a goal}\}$

$= \min\{\} = \infty \geq h(n_0)$

$h(n)=8 \qquad h(n')=4$

$n \xrightarrow{3} n' \xrightarrow{6} n''$ goal

$h^*(n) = 9 \geq 8 \quad h^*(n') = 6 \geq 4 \Rightarrow h$ is admissable

$h(n) = 8 > 7 = 3+4 = h(n') + c(n,a,n') \Rightarrow h$ is not consistent

In general: If n' is a child of n on a cheapest path from n to a goal s.t. $h^*(n') - h(n') > h^*(n) - h(n)$, then h is not consistent

Proof: $h(n) \geq h^*(n) - h^*(n') + h(n') = c(n,a,n') + h(n')$

Getting a heuristic:

- simplify the problem = relaxation
- get the optimal solution
- use this solution for the original problem

2.3. efficient calculation: $(\underline{0}83)(16)(2)(475)$
$$2 + 3 + 0 + 4 = 9$$

Know: $h_1 \leq h_2$ also: $h_1 \leq h_3$ (one Gaschnig move brings at most one tile to its correct place)

$\pi(n) \triangleq$ permutation representing node n, blank ones denoted by 0

$\pi = cyc\ \gamma_1 \dots \gamma_l \triangleq$ cyclic decomposition of permutation π

$|\gamma| \triangleq$ Length of cycle γ $\quad \#_3(\gamma) \triangleq$ number of needed Gaschnig moves for cycle

$h_3(n) \triangleq$ " for node

$h_3'(n) = \#(\gamma_1) + \dots + \#(\gamma_l)$ with $\pi(n) = cyc\ \gamma_1 \dots \gamma_l$

$$\#_3(\gamma) = \begin{cases} 0 & \text{if } |\gamma| = 1 \\ |\gamma| - 1 & \text{if } 0 \in \gamma \\ |\gamma| + 1 & \text{else} \end{cases} \qquad \#_1(\gamma) = \begin{cases} 0 & \text{if } |\gamma| = 1 \\ |\gamma| - 1 & \text{if } 0 \in \gamma \\ |\gamma| & \text{else} \end{cases}$$

$\#_1(\gamma)$ / $h_1(n)$ number of misplaced tiles

$\Rightarrow \#_1(\gamma) \leq \#_3(\gamma) \Rightarrow h_1(n) \leq h_3(n)$

example: interchange two or four pairs of adjacent tiles and maybe after that move the blank

2	1	3
8		4
6	7	5

$h_1: 4 \quad h_2: 4 \quad h_3: 6 \quad h^*: 16$

(2.4) initial state $(0,0)$ operations $e_3: (x,y) \mapsto (0,y)$ $\quad f_3: (x,y) \mapsto (3,y)$

a) reachable states $\quad e_4: (x,y) \mapsto (x,0)$ $\quad f_4: (x,y) \mapsto (x,4)$

state space $= \{0,3\} \times \{0,1,2,3,4\} \cup \{1,2\} \times \{0,4\}$ $\quad g_3: (x,y) \mapsto (x-2, y+2)$ $\quad g_4 \dots$

$\{0,1,2,3\} \times \{0,1,2,3,4\}$ is sufficient

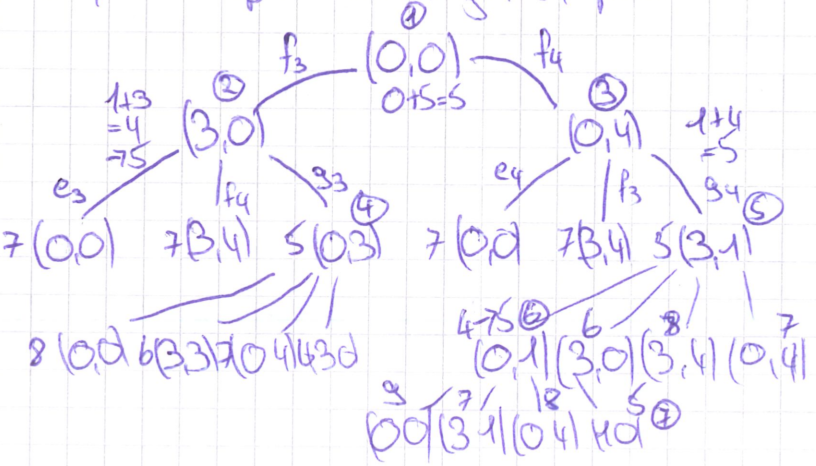
goal test: (x,y): iff $y = 2$ $\quad$ path cost = Length of past

b)

	state [n]	h(n)
	$(x,2)$	0
	$(0,0)$	5
	$(3,4)$	5
else	$(0,y)$	y
else	$(3,y)$	3
	else	1

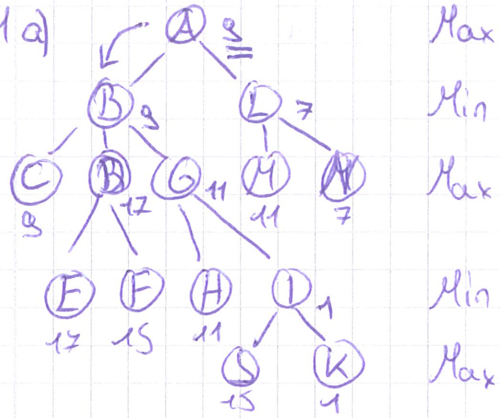
not consistent

→ path max correction

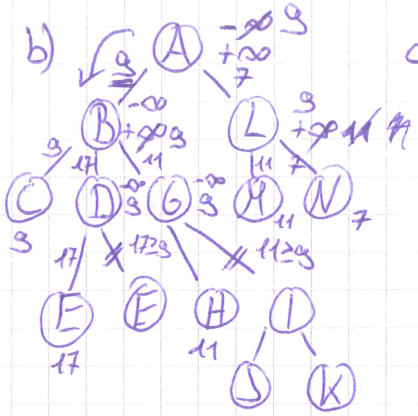


Complete

E3.1 a)

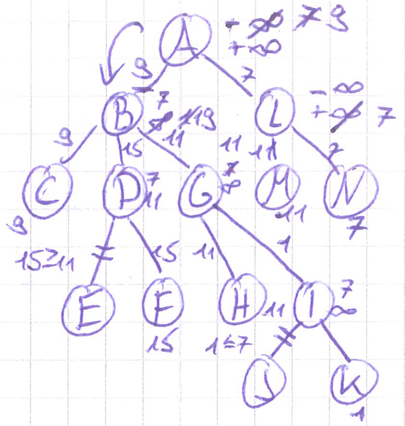


b)



pruned F, I, J, K

c)



pruned E, J

E3.2 a) function MAX-VALUE(state)

    if TERMINAL-TEST(state) ~~then return~~
        return UTILITY(state)
    v ← −∞
    for all a,s in SUCCESSOR(state)
        if Winner(s) = MAX
            v ← max(v, MAX-VALUE(s))
        else
            v ← max(v, MIN-VALUE(s)) (*)
    return v

(*) also includes when Winner(s) = ∅ (e.g. MIN has to follow suit)

b) L^2P

MAX [C10, S3, H4] [C8, H7, D6] : ∗ 0 ,

MAX: whose turn it is

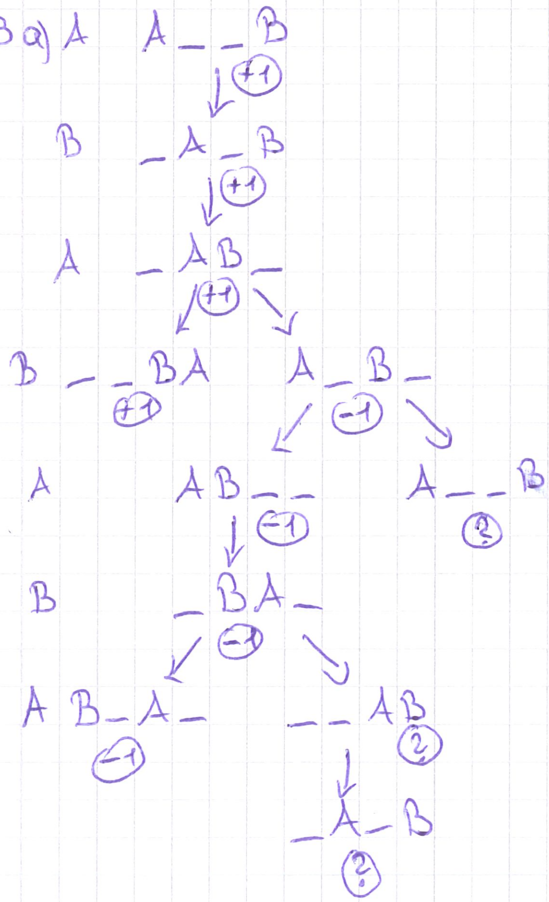
[C10, S3, H4]: MAX's hand

[C8, H7, D6]: MIN's Hand

∗: Card on table

0: tricks won by Max

E3.3 a)



c) standard minimax cannot handle question marks. If loop states would not have been considered as terminal states, then minimax would run into infinite path

fix: Keep track of loop states and define max and min for each non-empty subset of {+1, ? −1} (see ? as 0)

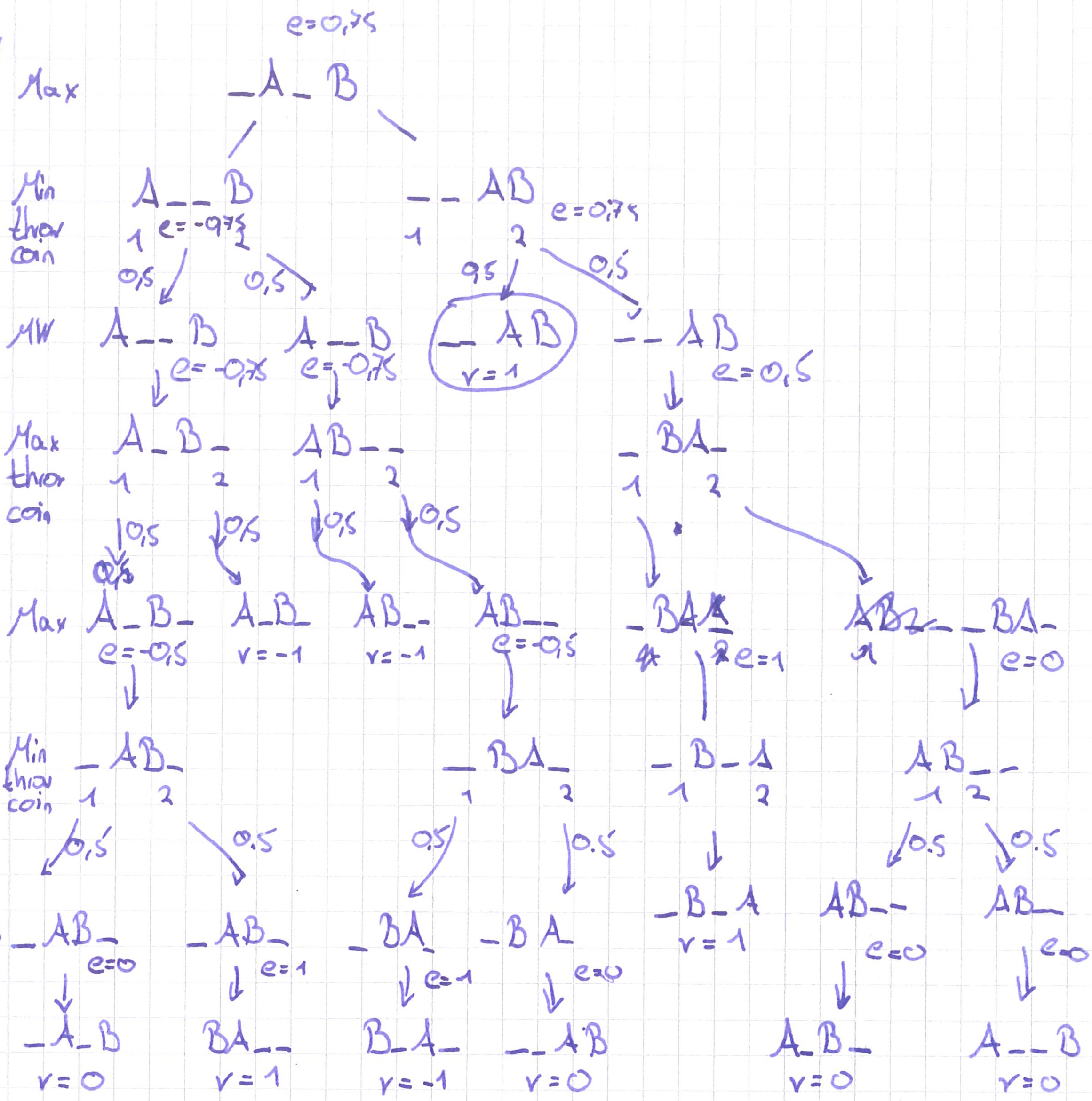
This will always work optimal for win/Lose games (ie when terminal states have values "+1" or "−1") because 0 represents a draw (or generally when there is a value for a draw)

d) $n > 2$ Claim: A wins if n is even or if n is odd and B moves first.

Strategy: move forward if possible, move backward otherwise

Proof: Let: Before A does the m-th move, A is at square a_m and B at b_m

3.4



4.1 a) α is valid, iff for all int. I, $I \models \alpha$ (def. of validity)

iff " I with $I \models true$, also $I \models \alpha$

(since all int. satisfy true)

iff $true \models \alpha$

b) Let α be an arbitrary sentence. Suppose $false \not\models \alpha$

Then there is some interpretation I s.t. $\underbrace{I \models false}_{\text{↯}}$, $I \not\models \alpha$

c) $\alpha \supset \beta$ is valid iff for all int. I, $I \models \alpha \supset \beta$

iff –"– $I \models \neg\alpha \vee \beta$ iff –"– $I \models \neg\alpha$ or $I \models \beta$

iff –"– $I \not\models \alpha \vee I \models \beta$ iff ... $(I \not\models \alpha$ and $I \models \beta)$ or $(I \not\models \alpha$ and $I \not\models \beta)$

or $(I \models \alpha$ and $I \models \beta)$ iff ... when $I \models \alpha$, then also $I \models \beta$

iff $\alpha \models \beta$

d) $\alpha \equiv \beta$ valid iff $(\alpha \supset \beta) \wedge (\beta \supset \alpha)$ valid iff $(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$ valid
iff $\forall I: I \models (\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$ iff $\forall I: (I \not\models \alpha$ or $I \models \beta)$
and $(I \not\models \beta$ or $I \models \alpha)$ iff $\forall I: (I \not\models \alpha$ and $I \not\models \beta$ or
$(I \not\models \alpha$ and $I \models \alpha)$ or $(I \models \beta$ and $I \not\models \beta)$ or $I \models \alpha$ and $I \models \beta$
iff $\forall I: I \models \alpha$ iff $I \models \beta$ iff $\{I \mid I \models \alpha\} = \{I \mid I \models \beta\}$
iff α and β are equivalent

e) $\alpha \wedge \neg\beta$ is unsatisfiable iff no I exists s.t. $I \models \alpha \wedge \neg\beta$
iff forall $I: I \not\models \alpha \wedge \neg\beta$ iff $\forall I: I \models \neg(\alpha \wedge \neg\beta)$
... $\forall I: I \models \alpha \supset \beta$ iff $\alpha \models \beta$ (part c)

4.2 Daughter(NS, parents(M)) (1)
$\neg$ Sister(NS, M) (2)
$\forall x \forall y$ Daughter$(x,y) \equiv$ Female$(x) \wedge$ parents$(x) = y \wedge y \neq x$ (3)
$\forall x \forall y$ Sister$(x,y) \equiv$ Female$(x) \wedge$ parents$(x) =$ parents$(y) \wedge y \neq x$ (4)
Claim: $KB = \{1,2,3,4\}$ $KB \models NS = M$
Let $I = \langle D, \phi \rangle$ be an int. with $I \models KB$ show: $I \models (M = NS)$
Let $a = \phi(NS)$, $b = \phi(M)$, $c = I\|parents(M)\| = \phi(parents)(\phi(M))$
$= \phi(parents)(b)$ (5)
With $I \models (1)$ we have $\langle a, c \rangle \in \phi(Daughter)$ (6)
$I \models (2)$ we have $\langle a, b \rangle \notin \phi(Sister)$ (7)
$I \models (3)$ $\langle \delta, \delta' \rangle \in \phi(Daughter)$ iff $\delta \in \phi(Female)$, $\delta' = \phi(parents)(\delta)$ $\delta \neq \delta'$
$\overset{(6)}{\Rightarrow}$ $a \in \phi(Female)$ ~~or $\phi(parents)(a) \neq \phi(parents)(b)$ or $a = b$~~
$c = \phi(parents)(a)$, $a \neq c$ (8)
$I \models (4)$: $\langle \delta, \delta' \rangle \in \phi(Sister)$ iff $\delta \in \phi(Female)$, $\phi(parents)(\delta) = \phi(parents)(\delta')$
$\delta \neq \delta'$
$\overset{(7)}{\Rightarrow}$ $\underbrace{a \notin \phi(Female)}_{\text{imp. (8)}}$ or $\underbrace{\phi(parents)(a) \neq \phi(parents)(b)}_{\text{imp. } \phi(parents)(a) \overset{(8)}{=} c \overset{(5)}{=} \phi(parents)(b)}$ or $a = b$

$\Rightarrow a = b \Rightarrow \phi(NS) = \phi(M) \Rightarrow I \models (M = NS)$ $\square$

4.3. Resolution

$S \models \alpha \Leftrightarrow$ for all int. I if $I \models S$, then $I \models \alpha$

$\Leftrightarrow$ f.a. int I, $I \not\models S \cup \{\neg\alpha\}$

$\Leftrightarrow$ $S \cup \{\neg\alpha\}$ onsatisfiable

$KB = \{\alpha_1, \dots \alpha_n\}$ $KB \models \alpha$ iff $\models (\alpha_1 \dots \alpha_n \supset \alpha)$

iff $KB \cup \{\neg\alpha\}$ onsatisfiable iff $KB \cup \{\neg\alpha\} \vdash$ false

Literals $\triangleq$ pos./neg. atoms

clauses $\triangleq$ disjonction of Literals

formulas $\triangleq$ sets of clauses (conjonction of clauses)

① convert to CNF

1. eliminate $\supset$ and $\equiv$ 2. push $\neg$ inwards 3. distribute $\vee$ over $\wedge$

4. simplify $(\alpha \vee \neg\alpha) \to$ true $\qquad (\alpha \wedge \beta) \vee \gamma \to (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$

a) $\{\} \vdash (P \supset (Q \supset P))$

$\{\} \cup \{\neg(P \supset (Q \supset P))\} \triangleq \neg(P \supset (Q \supset P)) = \neg(\neg P \vee (\neg Q \vee P))$

$= (\neg\neg P \wedge (\neg\neg Q \wedge \neg P)) = P \wedge Q \wedge \neg P$

$\triangleq \{[P], [Q], [\neg P]\}$

$[\,]$

b) $\{(P \supset (Q \supset R))\} \models ((P \supset Q) \supset (P \supset R))$

$\{(P \supset (Q \supset R))\} \cup \{\neg((P \supset Q) \supset (P \supset R))\}$

$\triangleq (P \supset (Q \supset R)) \wedge \neg((P \supset Q) \supset (P \supset R)) = (\neg P \vee (\neg Q \vee R)) \wedge \neg(\neg(\neg P \vee Q) \vee (\neg P \vee R))$

$= (\neg P \vee \neg Q \vee R) \wedge ((\neg\neg\neg P \vee \neg\neg Q) \wedge (\neg\neg P \wedge \neg R))$

$\triangleq \{[\neg P, \neg Q, R], [\neg P, Q], [P], [\neg R]\}$

$[\neg R]$ $[\neg P, \neg Q, R]$ $[\neg P, Q]$ $[P]$ $[\neg R]$

$[\neg P, \neg Q]$ $\qquad [Q]$

$[\neg P]$

$[\,]$

c) $\{(Q \supset P), (Q \supset \neg P)\} \models \neg Q$

$\{(Q \supset P, (Q \supset \neg P)\} \cup \{\neg\neg Q\} \stackrel{1}{=} (Q \supset P) \land (Q \supset \neg P) \land \neg\neg Q$

$= (\neg Q \lor P) \land (\neg Q \lor \neg P) \land (\neg\neg Q)$ $\quad [\neg Q, P]\ [\neg Q, \neg P]\ [Q]$

$\stackrel{1}{=} \{[\neg Q, P], [\neg Q, \neg P], [Q]\}$ $\quad [\neg Q] \quad [\,]$

4.4 a) $[P(f)]\ [Q(f')]\ [\sim P(z), \neg Q(z)]$

$z/f \quad z/f'$

$[\neg Q(f)] \quad [\sim P(f')]$

$I = \langle D, \phi \rangle \quad D = \{1, 2\} \quad \phi(P) = \{\langle 1 \rangle\} \quad \phi\, Q = \{\langle 2 \rangle\}$

$I \models \exists x\, P(x) \quad I \models \exists x\, Q(x) \quad I \not\models \exists x\, (P(x) \land Q(x))$

a) $\to$ wrong

b) $[P(x), Q(y)]\ [\sim P(f)]\ [\neg Q(f)]$

x/f

$[Q(y)] \ \underset{y/f}{\text{———}}\ [\,]$ $\quad \Rightarrow$ correct

c) split $\equiv$ to $\supset$ and $\subset$

$\supset \Rightarrow [P(x)]\ [Q(y)]\ [\neg P(f), \neg Q(f)]$ $\quad \subset \Rightarrow [P(x)]\ [Q(x)]\ [\neg P(f), \neg Q(f')]$

$x/f \quad y/f \quad x/f \quad x/f$

$[\,] \quad \neg P(f) \quad [\,] \quad [\neg P(f)]$

$\Rightarrow$ correct

d) $[\neg P(x), Q(x, g(x))]\ [P(g(f))]\ [\neg Q(g(f), g(g(f)))]$

$x \backslash g(f)\ [Q(g(f), g(g(f)))] - [\,]$ $\quad \Rightarrow$ correct

e) $[\neg P(f'(x)), P(f(x))]\ [\neg P(f(x)), Q(x, f(f'(x)))]\ [Q(x, f(x)), P(f'(x))]\ [\neg Q(f'', z)]$

$[\neg P(f'(x)), Q(x, f(f'(x)))] \quad - \quad [Q(x, f(f'(x))), Q(x, f(x))]$

$\Rightarrow$ correct $\quad z/f(f'(f'')) \quad x/f'' \quad z/f(f'')$

$[\,] - [Q(f'', f(f'(f'')))]$

f) wrong: $[P(x, f(x))]\ [\sim P(z, a)]\ [\sim P(f'(v), v)]$

$\Rightarrow$ not unifiable

term clash $\quad$ occur check failure

$DS = \{f(x), a\}$ $\quad DS = \cancel{\{x, f(x)\}}\ \{v, f(f'(v))\}$

Note: in general, not sufficient to use resolution to disprove a statement: may happen (in FOL) that we keep generating new clauses

Instead: counterexample! $I = \langle D, \phi \rangle \quad D = \{1, 2, 3\} \quad \phi(a) = 1 \quad \phi(P) = \{\langle 1, 2 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle\}$

$I \models \forall x \exists y\, P(x, y) \quad I \models \neg \exists z\, P(z, a) \quad I \not\models \exists y\, \forall x\, P(x, y)$

5.1. a) On(x,y) Clear(x) Clear(z)

~~Move~~ $\boxed{\text{Move}(x,y,z)}$

¬On(x,y) On(x,z) Clear(y) ¬Clear(z)

On(x,y) Clear(x)

$\boxed{\text{MoveToTable}(x,y)}$

¬On(x,y) On(x,T) Clear(y)

$\boxed{\text{Start}}$

On(A,T), On(B,T), On(C,T)
Clear(A), Clear(B), Clear(C)

On(A,B) On(B,C)

$\boxed{\text{Finish}}$

b) $\boxed{\text{Start}}$

On(A,T), On(B,T), On(C,T), Clear(A), Clear(B), Clear(C)

$y_1 = T$ On(B,y_1) Clear(B) Clear(C)

$\boxed{\text{Move}(B,y_1,C)}$

¬On(B,y_1) On(B,C) Clear(y_1) ¬Clear(C)

$y_2 = T$ On(A,y_2) Clear(A) Clear(B)

$\boxed{\text{Move}(A,y_2,B)}$

¬On(A,y_2) On(A,B), Clear(y_2) ¬Clear(B)

On(A,B) On(B,C)

$\boxed{\text{Finish}}$

c) Conflict: ¬Clear(B) / Clear(B)

Here only promotion (of Move(A,y_2,B)) is possible

Consistent: yes (no cyclic orderings)

Complete: yes (all preconditions satisfied, no more threats/conflicts)

5.2. a) P(toothache) = 0,108 + 0,012 + 0,016 + 0,064 = 0,2

b) $\mathbb{P}$(Cavity) = ⟨P(cavity), P(¬cavity)⟩

= ⟨0,2, 0,8⟩

c) $\mathbb{P}$(Toothache | cavity) = ⟨P(toothache | cavity), P(¬toothache, cavity)⟩

$= \langle \frac{P(\text{toothache} \wedge \text{cavity})}{P(\text{cavity})}, \frac{P(\neg\text{toothache} \wedge \text{cavity})}{P(\text{cavity})} \rangle$

$= \langle \frac{0{,}108 + 0{,}012}{0{,}2}, \frac{0{,}072 + 0{,}008}{0{,}2} \rangle = \langle 0{,}6, 0{,}4 \rangle$

d) P(toothache ∨ cavity) = 0.108 + 0,012 + 0,072 + 0,016 + 0,064 + 0,144

= 0,416

$\mathbb{P}$(Cavity | toothache ∨ catch) = ⟨P(cavity | toothache ∨ catch), P(¬cavity | ...)⟩

$= \langle \frac{P(A \wedge B)}{P(B)}, \frac{P(\neg A \wedge B)}{P(B)} \rangle = \langle \frac{0.108 + 0.012 + 0.072}{0.416}, \frac{0.016 + 0.064 + 0.144}{0.416} \rangle$

$= \langle 0.4615..., 0.5384... \rangle$

5.3 Claim: $P(A) = P(A|B_1)\,P(B_1) + \ldots + P(A|B_n) \cdot P(B_n)$

with $P(B_1 \vee \ldots \vee B_n) = 1$ and $P(B_i \wedge B_j) = 0$ for $i \neq j$

Proof: $P(A|B_1) \cdot P(B_1) + \ldots + P(A|B_n) \cdot P(B_n)$

$\overset{\text{def.}}{=} P(A \wedge B_1) + \ldots + P(A \wedge B_n)$

$\overset{*}{=} P(A \wedge [B_1 \vee \ldots \vee B_n]) = P(A)$

$*$ Proof by induction on k

IH: $P(A \wedge B_1) + \ldots + P(A \wedge B_k) = P(A \wedge [B_1 \vee \ldots \vee B_k])$

IB: $k = 1 \checkmark \quad k = 0 \checkmark$

IS: $P(A \wedge B_1) + \ldots + P(A \wedge B_{k+1}) \overset{IH}{=} P(A \wedge [B_1 \vee \ldots \vee B_k]) + P(A \wedge B_{k+1})$

$= P([A \wedge [B_1 \vee \ldots \vee B_k]] \vee [A \wedge B_{k+1}]) + P([A \wedge [B_1 \vee \ldots \vee B_k]] \wedge [A \wedge B_{k+1}])$

$= P(A \wedge [B_1 \vee \ldots \vee B_{k+1}]) + \underbrace{P(A \wedge [B_1 \vee \ldots \vee B_k] \wedge B_{k+1})}_{** = 0}$

$**$ Proof by induction on k

IH: $P([B_1 \vee \ldots \vee B_k] \wedge B_l) = 0$ for $l > k$

IB: $k = 1 \checkmark \;|\; k = 0 \quad P(\text{false} \wedge B_l) = 0$

IS $P([B_1 \vee \ldots \vee B_{k+1}] \wedge B_l)$ with $l > k+1$

$= P([B_1 \vee \ldots \vee B_k] \wedge B_l \vee [B_{k+1} \wedge B_l])$

$\overset{IH}{=} P(B_{k+1} \wedge B_l) = 0$

5.4 a) $P(Q_i|W) = 0.95$ $P(Q$ for $i = 1,2,3$ $P(W)$ $0,8$

$P(Q_1|\neg W) = 0,3 \quad P(Q_2|\neg W) = 0,5 \quad P(Q_3|\neg W) = 0.1$

b) $P(W|Q_1) = \dfrac{P(Q_1|W) \cdot P(W)}{P(Q_1)} = \dfrac{P(Q_1|W) \cdot P(W)}{P(Q_1|W) \cdot P(W) + P(Q_1|\neg W) \cdot P(\neg W)} = \dfrac{0.95 \cdot 0.8}{0.95 \cdot 0.8 + 0.3 \cdot 0.2} \approx 0.927$

$P(Q_1, Q_2|\neg W) \underset{\uparrow}{=} P(Q_1|\neg W) \cdot P(Q_2|\neg W) = \cancel{0.95 \cdot 0.95 = 0.9025}$

$\cancel{0.05 \cdot 0.05}$ $= 0.15$

$0.3 \cdot 0.5$

Q_i are conditionally independent given W

$P(Q_3|Q_1, Q_2, W) = P(Q_3|W) = 0.95$

Bayesian update: c) $P(W|Q_1, Q_2, \neg Q_3) = \alpha \cdot P(W) \cdot P(Q_1|W) \cdot P(Q_2|W) \cdot \underset{1 - P(Q_3|W)}{P(\neg Q_3|W)}$

$= \alpha \cdot 0.8 \cdot 0.95 \cdot 0.95 \cdot (1 - 0.95) \approx \alpha \cdot 0.0361 \approx 0.572$

$P(\neg W|Q_1, Q_2, \neg Q_3) = \alpha \cdot \ldots = \alpha \cdot 0.0270 \Rightarrow \alpha \cdot 0.0631 = 1 \Rightarrow \alpha = 15.847$

d) $P(W|Q_1, \neg Q_2, \neg Q_3) = \beta \cdot 0.019 \quad \beta = \frac{1}{0.0283} = 0.066$

e) Because this means that the Q_i are conditionally independent given W

6.1

a) $P(W \land \neg L \land R \land S=spring) = P(W \mid \neg L \land R \land S=spring) \cdot P(\neg L \mid R \land S=spring)$

$\cdot P(R \mid Spring=spring) \cdot P(S=spring)$

$= P(W \mid \neg L \land R) \cdot P(\neg L \mid S=spring) \cdot P(R \mid S=spring) \cdot P(S=spring)$

$= 0.95 \cdot (1-0.15) \cdot 0.45 \cdot 0.25 \approx 0.0908$

b) $P(S=winter \mid \neg R \land \neg L) = \dfrac{P(S=winter \land \neg R \land \neg L)}{P(\neg R \land \neg L)}$

$(**) = \dfrac{P(\neg R \mid S=winter) \cdot P(\neg L \mid S=winter) \cdot P(S=winter)}{(*) \sum_{x \in Season} P(\neg R \land \neg L \land S=x) = \sum_{x \in Season} P(\neg R \mid S=x) \cdot P(\neg L \mid S=x) \cdot P(S=x)}$

$(*) = (1-0.45)(1-0.15) \cdot 0.25$

$+ (1-0.35)(1-0.05) \cdot 0.25$

$+ (1-0.35)(1-0.05) \cdot 0.25$

$+ (1-0.25)(1-0) \cdot 0.25 = 0.62$

$(**) = \dfrac{(1-0.2) \cdot (1-0) \cdot 0.25}{0.62} \approx 0.3226$

c) $P(R \mid W \land S=sommer) = \dfrac{P(R \land W \land S=Sommer)}{P(W \land S=Sommer)}$

$= \dfrac{P(R \land W \land S=Sommer \land L) + P(R \land W \land S=sommer \land \neg L) \; (*)}{\cancel{\alpha} \sum_{r \in R} \sum_{l \in L} P(W \land S=sommer \land R=r \land L=l) \; (**)}$

$(*) = \cancel{\alpha} \cdot [P(W \mid R \land L) \cdot P(R \mid s=sommer) \cdot P(L \mid S=sommer) \cdot P(s=sommer)$

$+ P(W \mid R \land \neg L) \cdot P(R \mid s=sommer) \cdot P(\neg L \mid S=sommer) \cdot P(S=sommer)]$

$= \cancel{\alpha} \cdot [0.95 \cdot 0.15 \cdot 0.3 \cdot 0.25 + 0.95 \cdot 0.15 \cdot (1-0.3) \cdot 0.25]$

$= \cancel{\alpha} \cdot 0.035625$

$(**) = \sum_{r \in R, l \in L} P(W \land S=sommer \land R=r \land L=l)$

$= P(W \mid \neg R \land L) \cdot P(\neg R \mid S=sommer) \cdot P(L \mid s=sommer) \cdot P(S=sommer)$

$+ P(W \mid \neg R \land \neg L) \cdot \; \ldots$

$+ P(W \mid R \land L) \cdot \; \ldots$

$+ P(W \mid R \land \neg L) \cdot \; \ldots \; = 0.1004375$

$\Rightarrow \dfrac{0.035625}{0.1004375} \approx \cancel{\alpha}\, 0.3547$

6.2

a) $X = \{U\}$ $Y = \{V\}$ $E = \{Z\}$

3. case applies $\boxed{U} \to \underline{\underline{W}} \leftarrow \boxed{V}$ and $\boxed{U} \to W \to \underline{\underline{Y}} \leftarrow X \leftarrow \boxed{V}$

$\Rightarrow$ can be shown using d-seperation ✓

b) $X = \{U, W\}$ $Y = \{X, Z\}$ $E = \{Y\}$

$\boxed{U \to W} \leftarrow V \to \boxed{X \to Z}$ none

$\boxed{U \to W} \to \boxed{Y} \leftarrow \boxed{X \to Z}$ none

$\Rightarrow$ cannot show independence with d-seperation ✗

c) $X = \{U, W\}$ $Y = \{X, Z\}$ $E = \{V\}$

$\boxed{U \to W} \leftarrow \boxed{V} \to \boxed{X \to Z}$ 2. case applies

$\boxed{U \to W} \to \underline{\underline{Y}} \leftarrow \boxed{X \to Z}$ 3 " ✓

d) $X = \{W\}$ $Y = \{X\}$ $E = \{U\}$

$\boxed{W} \leftarrow \boxed{V} \to \boxed{X}$ none

✗

$\boxed{W} \to \underline{Y} \leftarrow \boxed{X}$ 3)

e) $X = \{U\}$ $Y = \{V\}$ $E = \{\}$

$\boxed{U} \to \underline{\underline{W}} \leftarrow \boxed{V}$ 3) ✓

$\boxed{U} \to W \to \underline{\underline{Y}} \leftarrow X \leftarrow \boxed{V}$ 3)

f) $X = \{U\}$ $Y = \{Z\}$ $E = \emptyset$

$\boxed{U} \to \underline{\underline{W}} \leftarrow V \to X \to \boxed{Z}$ ~~none~~ 3) ~~✗~~

$\boxed{U} \to W \to \underline{\underline{Y}} \leftarrow X \to \boxed{Z}$ 3) ✓

6.3

$I(P(v_1), \dots, P(v_k)) = \sum_{i=1}^{k} -P(v_i) \cdot \log_2 P(v_i)$ (information content)

$\text{Remainder}(A) = \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} \cdot I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$

$\text{Gain}(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \text{Remainder}(A)$

$p = 3$; $n = 5$ $\quad I(3/8, 5/8) = -3/8 \cdot \log_2 3/8 - 5/8 \cdot \log_2 5/8 \approx 0.9544$

A_1 = country: 1) Italy: $n_1 = 2$ $p_1 = 0$ $I(1, 0) = 0$

2) Austria: $n_2 = \cancel{2}1$ $p_2 = \cancel{1}2$ $I(1/3, 2/3) \approx 0.918$

3) Spain: $n_3 = 2$ $p_3 = 1$ $I(2/3, 1/3) \approx 0.918$

$R(A_1) = 1/4 \cdot 0 + 3/8 \cdot 0.918 + 3/8 \cdot 0.918 \approx 0.689$ $\quad G(A_1) \approx 0.2654$

A_2 = season: 1) sommer: $n_1=3$ $p_1=0$ $I(1,0)=0$

2) winter: $n_1=2$ $p_2=3$ $I(2/5, 3/5) \approx 0.971$

$R(A_2) = 3/8 \cdot 0 + 5/8 \cdot 0.971 \approx 0.607$ $\quad Gain(A_2) = 0.9544 - 0.607 \approx 0.3474$

A_3 = type: 1) repose $n_1=2$ $p_1=2$ $I(1/2, 1/2)=1$

2) sports $n_2=3$ $p_2=0$ $I(1,0)=0$

3) culture $n_3=0$ $p_3=1$ $I(0,1)=0$

$R(A_3) = 4/8 \cdot 1 + 3/8 \cdot 0 + 1/8 \cdot 0 = 1/2$ $\quad Gain(A_3) = \underline{0.4544} \Rightarrow$ best

A_4 = weeks: 1) 1: $n_1=2$ $p_1=1$ $I(2/3, 1/3) \approx 0.918$

2) 2: $n_2=2$ $p_2=1$ "

3) 3: $n_3=1$ $p_3=1$ $I(1/2, 1/2)=1$

$R(A_4) = 3/8 \cdot 0.918 + 3/8 \cdot 0.918 + 1/4 \cdot 1 \approx 0.9385$ $\quad Gain(A_4)\ 0.0159$

$A_3 \Rightarrow$ repose $I(1/2, 1/2)=1$

A_2 = season 1) sommer: $n_1=2$ $p_1=0$ $I(1,0)=0$

2) winter: $n_2=0$ $p_2=2$ $I(0,1)=0$

$R(A_2) = 1/2 \cdot 0 + 1/2 \cdot 0 = 0$ $\quad Gain(A_2) = \underline{\underline{1}} \Rightarrow$ best

A_1 = country 1) Italy $n_1=1$ $p_1=0$ $I(1,0)=0$

2) Austria $n_2=0$ $p_2=1$ $I(0,1)=0$

3) Spain $n_3=1$ $p_3=1$ $I(1/2, 1/2)=1$

$R(A_1) = 1/4 \cdot 0 + 1/4 \cdot 0 + 2/4 \cdot 1 = 0.5$ $\quad Gain(A_1) = 0.5$

A_4 = weeks 1) 2 $n_1=1$ $p_1=1$ $I(1/2, 1/2)=1$

2) 3 $n_2=1$ $p_2=1$ $I(1/2, 1/2)=1$

$R(A_4) = 1/2 \cdot 1 + 1/2 \cdot 1 = 1$ $\quad Gain(A_4)=0$

6.4. $I_1 \xrightarrow{w_1} \dots I_m \xrightarrow{w_n} O \rightarrow O = \text{step}_0\left(\sum_{j=1}^{n} I_j \cdot W_j\right)$ update: $W_j + \alpha \cdot I_j \cdot E$

a)

example	I_A	I_B	I_r	$I_\#$	I_*	I_g	I_n	I_b	T
1	0	0	1	2	1	1	0	0	1
2	0	1	0	3	0	0	0	1	0
3	0	1	0	1	0	0	1	0	0
4	1	0	0	2	1	0	0	1	1
5	1	0	0	1	0	1	0	0	1
6	0	0	1	3	1	0	0	1	1
7	0	1	0	1	0	1	0	0	1
8	0	0	1	2	0	0	0	1	0

b)

attribute	brand	persons	attribute	parking	gender
encoding	distributed	local	local	distr.	local

c)

	example	O	E	W_A	W_B	W_r	$W_\#$	W_*	W_g	W_n	W_b
Init				+1	+1	+1	+1	+1	+1	+1	+1
	1	1	0								
	2 ⋮										
	4	0	+1	+3	-1		-1	+3			
E_1	5	1	0								
	6	1	0								
	7	0	+1		+1		+1	[illegible]	+~~4~~3		
	8	1	-1			-1	-3				-1
	1	0	+1			+1	+1	+5	+5		
	2	1	-1		-1	[illegible]	-5				-3
	3	0	0								
	4	0	+1	+5			-1	+7			-1
E_2	5	1	0								
	6	1	0								
	7	1	0								
	8	0	0								

no changes in E_3 => done

Q & A

search:

Two possibilities for applying goal test:

(A) after generating a node

(B) when selecting a node for expansion

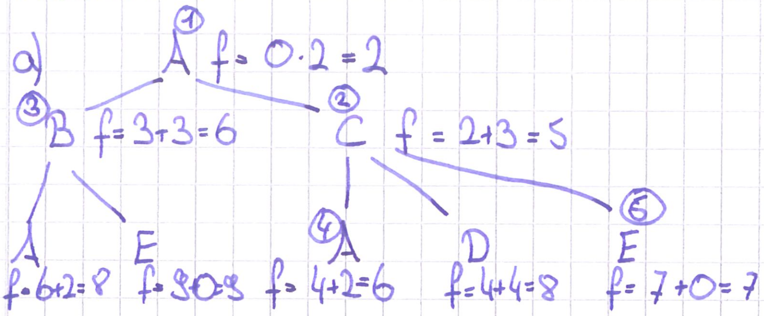
BFS uses (B) IDS uses (A)

(B) is not necessary for BFS (would work ~~f~~ with (A) as well), only for UCS (of which BFS is a special case for uniform costs) in to be optimal

Still (A)-IDS is not much worse than A-BFS in terms of time

Problem 2

a)



b) A-C-D-E is optimal (costs 5)

c) heuristic function h is not admissable, i.e. it overestimates

$h(D) = 4$, but D-E has cost 1

e.g. let $h'(n) = \begin{cases} 1, & n = D \\ h(n) & \text{otherwise} \end{cases}$ or $h' = 0$ (A^* becomes UCS)

Problem 4

c) $P(D,E \mid \neg C, B) = \dfrac{P(B,\neg C,D,E)}{P(B,\neg C)} = \dfrac{P(A,B,\neg C,D,E) + P(\neg A,B,\neg C,D,E)}{P(B,\neg C)}$

~~$= \dfrac{P(D \mid A,\neg C,E) \cdot P(E \mid \neg C) \cdot P(\neg C \mid B) \cdot P(B) \cdot P(A) + P(D \mid \neg A,\neg C,E) \cdot P(E \mid \neg C) \cdot P(\neg C \mid B) \cdot P(B) \cdot P(\neg A)}{P(\neg C \mid B) \cdot P(B)}$~~

$= P(D \mid A,\neg C,E) \cdot P(E \mid \neg C) \cdot P(A) + P(D \mid \neg A,\neg C,E) \cdot P(E \mid \neg C) \cdot P(\neg A)$

$= 0.8 \cdot 0.75 \cdot P(A) + 0.4 \cdot 0.75 \cdot (1 - P(A))$

$= 0.6 \cdot P(A) + 0.3(1-P(A)) = 0.3 \cdot P(A) + 0.3 \overset{!}{\geq} 0.5$

$\Leftrightarrow 0.3\,P(A) \geq 0.2 \Leftrightarrow P(A) \geq 2/3$

a) $P(\neg B \mid E) = \dfrac{P(\neg B,E)}{P(E)} = \dfrac{P(\neg B,C,E) + P(\neg B,\neg C,E)}{P(B,C,E) + P(B,\neg C,E) + P(\neg B,\neg C,E) + P(\neg B,C,E)}$

$= \dfrac{P(E \mid C) \cdot P(C \mid \neg B) \cdot P(\neg B) + P(E \mid \neg C) \cdot P(\neg C \mid \neg B) \cdot P(\neg B)}{P(E \mid C) \cdot P(C \mid \neg B) \cdot P(\neg B) + P(E \mid \neg C) \cdot P(\neg C \mid B) \cdot P(B) + \text{---}\,''\,\text{---}}$

$= \dfrac{0.25 \cdot 0.6 \cdot 0.4 + 0.75 \cdot 0.4 \cdot 0.4}{0.25 \cdot 0.9 \cdot 0.6 + 0.75 \cdot 0.1 \cdot 0.6 + \text{---}\,''\,\text{---}}$

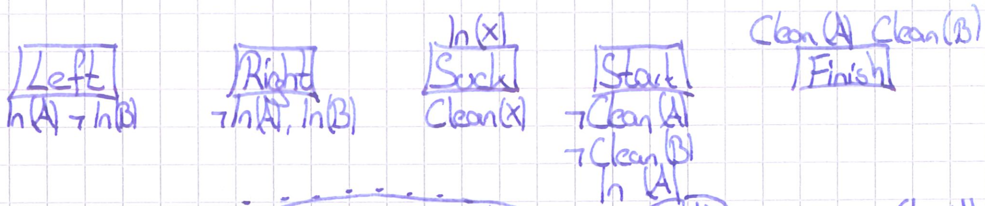
$= \dfrac{0.06 + 0.12}{0.135 + 0.045 + 0.06 + 0.12} = \dfrac{0.18}{0.36} = \dfrac{1}{2}$

b) $B \to \underline{C} \to D$ (condition 1) $\mathcal{E} = \{C\}$ (or $\mathcal{E} = \{C,E\}$)

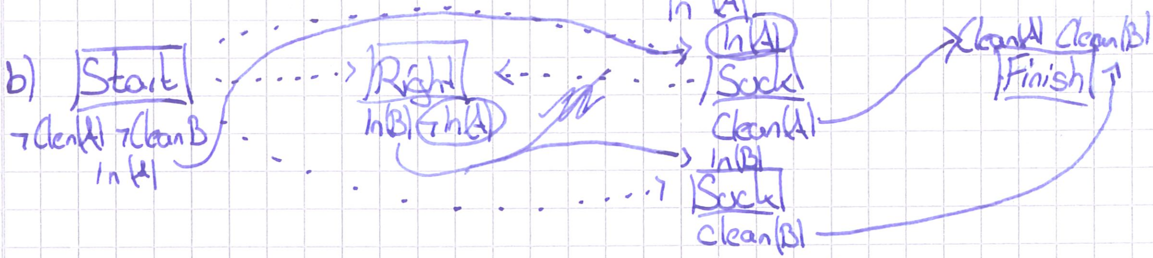
$B \to \underline{C} \to E \to D$ (condition 1)

Problem 7

a)



b)

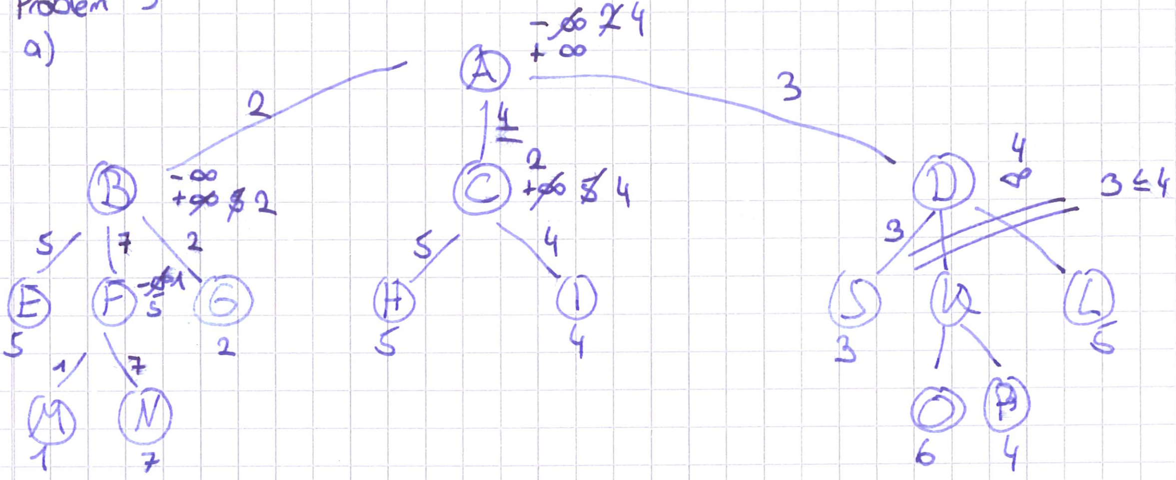


Note: To obtain a threat, causal link $\boxed{\text{Start}} \xrightarrow{In(A)} \boxed{\text{Suck}}$ is required (problem was not precise here)

Here only Suck ---> Right is possible for resolving the conflict
Right --X--> Start

Problem 3

a)



b)