1. (a) • Diagnostic reasoning : from symptom to cause
• Probabilistic reasoning : from cause to symptom
• Intercausal reasoning : mutual causes of a common effect
• Combined reasoning : variable descendent and parent of some observed variables

(b)

$P(C|B) = 0.0014$

$P(C) = 0.0002$

$P(B) = 0.01$

$$P(B|C) = \frac{P(C|B)\,P(B)}{P(C)} = \frac{0.0014 \times 0.01}{0.0002} = 0.07 = 7\%$$

(c) $P(A|F) = 0.93$ $\quad P(B|F) = 0.7$
$P(A|\neg F) = 0.16$ $\quad P(B|\neg F) = 0.01$

$P(F) = 0.002$

• $$P(F|A) = \frac{P(A|F)\,P(F)}{P(A)} = \frac{P(A|F)\,P(F)}{P(A|F) \times P(F) + P(A|\neg F) \times P(\neg F)}$$

$$= \frac{0.93 \times 0.002}{0.93 \times 0.002 + 0.16 \times 0.998}$$

$$\approx 0.01151$$

• $$P(F|B) = \frac{P(B|F)\,P(F)}{P(B|F) \times P(F) + P(B|\neg F) \times P(\neg F)} = \frac{0.7 \times 0.002}{0.7 \times 0.002 + 0.01 \times 0.998}$$

$$\approx 0.1230$$

Yes B $\quad P(F|B) > P(F|A)$

$$P(D \wedge H \wedge FD \wedge MS \wedge PO) + P(D \wedge H \wedge FD \wedge MS \wedge \neg PO)$$

(d) $P(D \mid H \wedge FD) = P(D \wedge H \wedge FD \wedge MS \wedge PO) + P(D \wedge H \wedge FD \wedge \neg MS \wedge PO)$

$= P(D) \times P(H \mid PO) \times P(FD \mid PO) \times P(PO \mid D \wedge MS) \times P(MS) \ldots + \ldots$

$+ \ldots$

$+ \ldots$

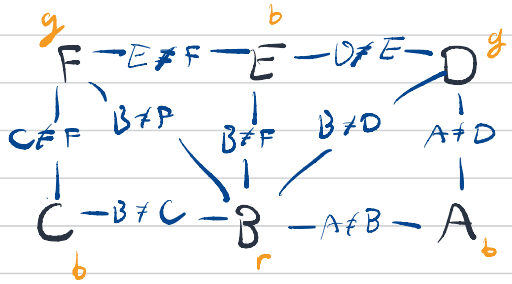
≈ 0.003

2. (a) MRV - pick variable with fewest compatible variables

DH - pick variable with most constraints on remaining variables

LCV - pick value with least constraints with remaining variable values.

(b) (i) all variables have domain {red, green, blue}



(ii)

A	B	C	D	E	F	Step
r, g, b	(r) g, b	r, g, b	r, g, b	r, g, b	r, g, b	MCV = any, DH = B, LCV = any: r
g, b	r	g, b	(g), b	g, b	g, b	MCV = any, DH = (D) E/F, LCV = any: g
b	r	g, b	g	(b)	g, b	MCV = A/E, DH = E, LCV = b
b	r	g, b	g	b	(g)	MCV = A/F, DH = F, LCV = g
(b)	r	b	g	b	g	MCV = A/C, DH = (A)/C, LCV = b
b	r	(b)	g	b	g	MCV = C, LCV = b
b	r	b	g	b	g	

(c) • Cutset condition on {M3} ← source of all cycles
• Then have 1 tree with |M| instances = 3 instances

$$d^c (n-c)\, d^2 = 3^1 (13-1)\, 3^2 = 324$$

$$d^n = 3^{13} = 1,590,323$$

(d)

• Encode the parameters into a string:

→ For every light, add to string $x_1 x_2 x_3 y_1 y_2 y_3$

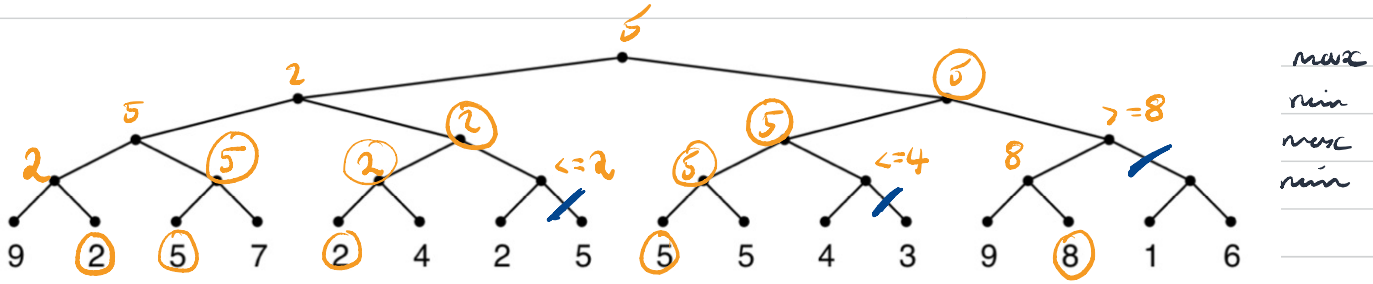
where x = green time, y = red time, in seconds.

• Run genetic algorithm using this string ← will have length $6 \times 50 = 300$

• Decode using reverse of encoding

Explain how genetic algo's work (with buzzwords)

3. (a) Alpha-beta pruning allows us to not search branches of a min / max node that will not change the utility due to being greater / less than the intermediate value we already have.



Move right, expect 5.

(b) Add a chance node who's expected utility is equal to the sum of minimax of the subtrees multiplied with the probability of taking that subtree.

(c) (i) Pick a cut point such that the utility at this depth varies little if a deeper cut point was chosen.

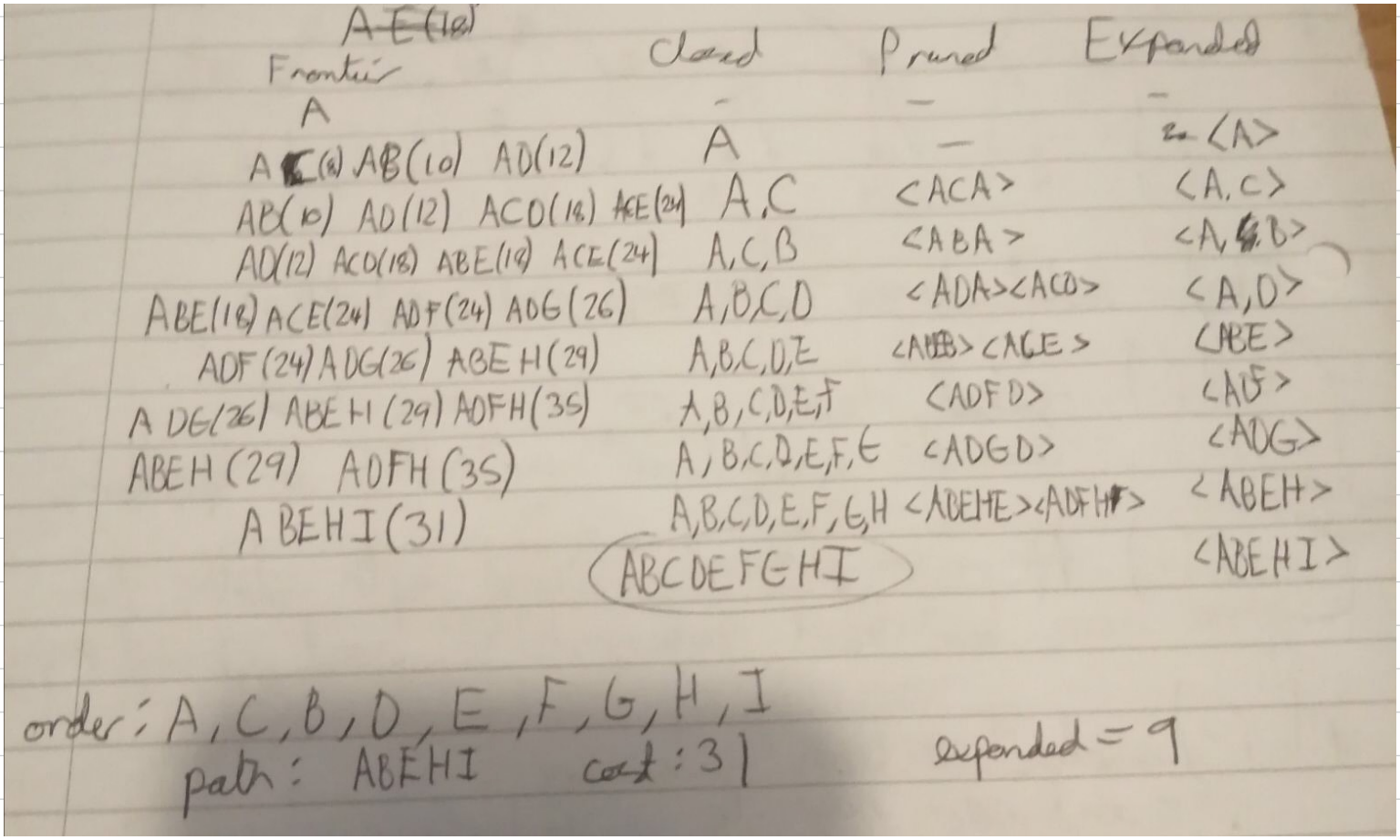
(ii) An unavoidable damaging move from an opponent may seem to be 'avoided' with a stalling move. ?!?

(d)(i) 1. Select an open precondition
2. Choose an operation or add a new operation that achieves this precondition.
3. Add the causal link from the operation to the precondition.
4. Resolve any clobbering.

(ii) • Clobbering occurs when an operations post-condition threatens a causal link.
• This means we must order the operation.
• We can choose to promote the operation to happen after or demote to happen before the operation involved in the causal link.

(iii) • A new type of operation that can decide which branch to take given a condition
• Useful since we may not know all info at the planning stage or want to account for uncertainty.

4. (a) (i) Uniform-cost search GRAPH SEARCH

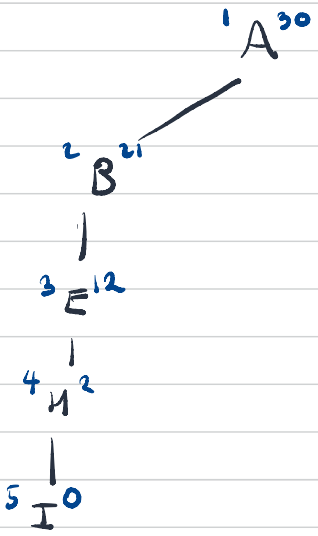


order: A, C, B, D, E, F, G, H, I

path: ABEHI cost: 31

expanded = 9

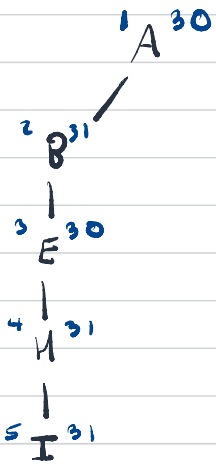
(iii) Best-First-Search



Step	Queue	Closed List	Action
1	$\langle A \rangle_{30}$		expand $\langle A \rangle_{30}$
2	$\langle AB \rangle_{21}$, $\langle AD \rangle_{24}$, $\langle AC \rangle_{29}$	A	expand $\langle AB \rangle_{21}$
3	$\langle ABE \rangle_{12}$, $\langle AD \rangle_{24}$, $\langle AC \rangle_{29}$	A, B	expand $\langle ABE \rangle_{12}$
4	$\langle ABEM \rangle_{2}$, $\langle AD \rangle_{24}$, $\langle AC \rangle_{29}$, $\langle ABEC \rangle_{29}$	A, B, E	expand $\langle ABEM \rangle_{2}$
5	$\langle ABEMI \rangle_{0}$, $\langle ABEMF \rangle_{21}$, $\langle AD \rangle_{24}$, $\langle AC \rangle_{29}$, $\langle ABEC \rangle_{29}$	A, B, E, M	expand $\langle ABEMI \rangle_{0}$
6	goal expanded		

- 5 nodes expanded
- Route: $\langle ABEMI \rangle_{0}$, cost: 31

(iii) A* Search



Step	Queue	Closed List	Action
1	$\langle A \rangle_{30}$		expand $\langle A \rangle$
2	$\langle AB \rangle_{31}^{10+21}$, $\langle AD \rangle_{36}$, $\langle AC \rangle_{37}$	A	expand $\langle AB \rangle$
3	$\langle ABE \rangle_{30}$, $\langle AD \rangle_{36}$, $\langle AC \rangle_{37}$	A, B	expand $\langle ABE \rangle$
4	$\langle ABEH \rangle_{31}$, $\langle AD \rangle_{36}$, $\langle AC \rangle_{37}$, $\langle ABEC \rangle_{63}$	A, B, E	expand $\langle ABEH \rangle$
5	$\langle ABEHI \rangle_{31}$, $\langle AD \rangle_{36}$, $\langle AC \rangle_{37}$, $\langle ABEHF \rangle_{32}$, $\langle ABEC \rangle_{63}$	A, B, E, H	expand $\langle ABEHI \rangle$
6	goal expanded		

- 5 nodes expanded
- Route $\langle ABEHI \rangle_{31}$, cost: 31

(b) No longer admissible but would still find the optimal path in our example.

(c) Why is A^* admissible?

- If a path p to a goal is selected from a frontier, can there be a shorter path to a goal?
- Suppose path p' is on the frontier. Because p was chosen before p', and $h(p) = 0$:

$$cost(p) \leq cost(p') + h(p')$$

- Because h is an underestimate:

$$cost(p') + h(p') \leq cost(p'')$$

for any path p'' to a goal that extends p'

- So $cost(p) \leq cost(p'')$ for any other path p'' to a goal.

(d) - DFS with max-depth, increase the depth and run again
- Becomes a BFS with linear space.

5. (a) • Forward chaining keeps track of what it knows and adds to this by seeing what it could derive with what it knows.

• Backward chaining starts with what we want to derive and finds out how we could derive this, and continues going back until we are only left with facts.

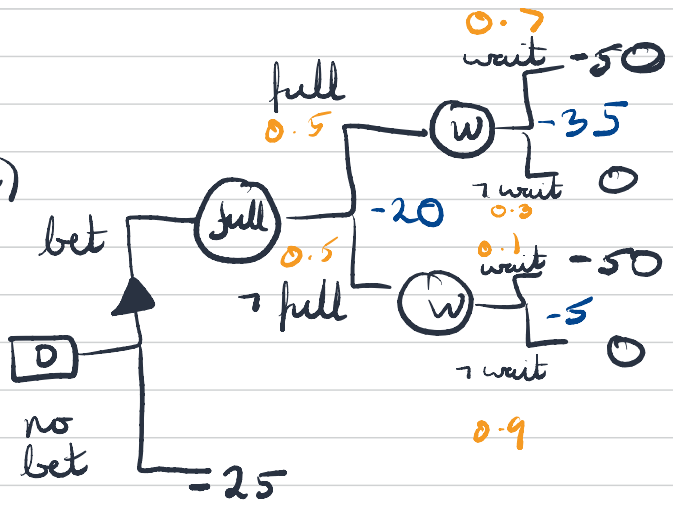
e.g. fact = {A} rules = {A → B, A → C, B ∧ C → D}
want to derive D

Forward Chaining	Backward Chaining
• A	• D
• A → B	• B ∧ C → D
• A → C	• A → B
• B ∧ C → D	• A → C
• All rules applied and derived D.	• Only left with facts

(b) (i) Conflict resolution is handling non-syntactic errors such as incorrect derivations, no answer produced, stuck in a loop etc.

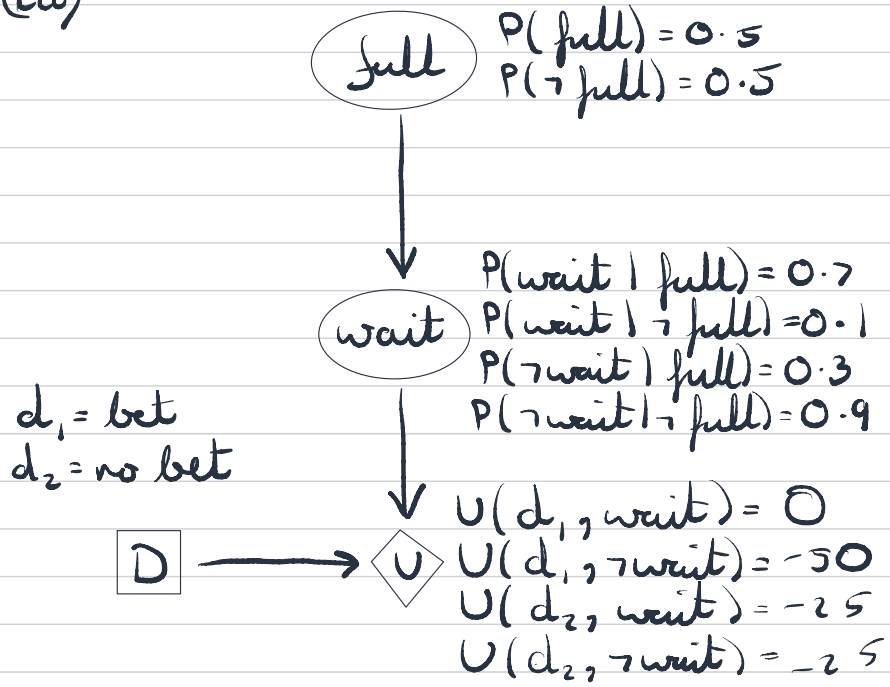
(ii) • Specificity: fire the most specific rule → more relevant to current situation
• Refractoriness: only allows rules to fire once → prevents loops on the same data
• Recency: fire rule that uses most recently derived data → focuses on relevant data first

(c) (i)



(ii) Should bet:

(iii)



(iv) Assuming review availability is a probability

- Would need to know: P(full | review available)
  P(full | ¬review available)

  P(review available)