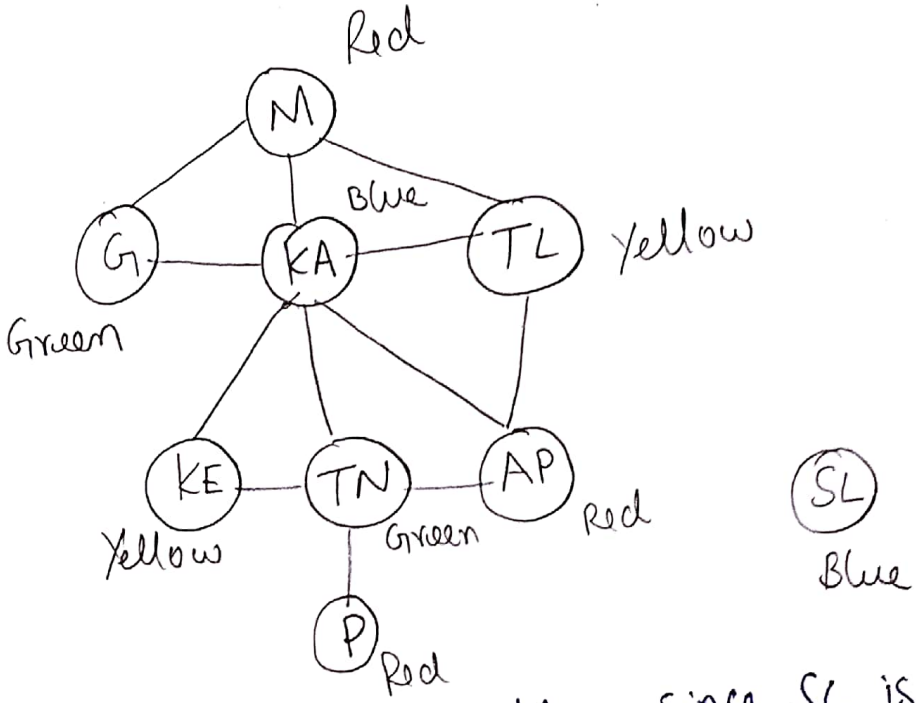
Kevin thomas
1001544593

CSE 5360-001

Assignment-5

1 a)



1) We can simplify this problem since SL is not connected to any other nodes, it is an independent subproblem.

2) Therefore we can solve the subproblem (M, G, KA, TL, KE, TN, AP, P) and SL subproblem independently.

3) We can combine the solution of these two subproblem to find the final solution.

b) MRV : 1st Subproblem

Level 0: all nodes → 4 Values (R, G, R, Y)

Level 1 : M → Pick red

G, KA, TL → 3 possible Values (G, B, Y)

KE, TN, AP, P → 4 possible Values (R, G, B, Y)

Level 2:

M → Red

G → Green

KA → 2 possible values (B, Y)

TL → 3 possible values (G, B, Y)

KE, TN, AP, P → 4 possible values (R, G, B, Y)

Level 3:

M → Red

G → Green

KA → Blue (Picking randomly)

TL → 2 possible values (G, Y)

KE, TN, AP → 3 possible values (R, G, Y)

P → 4 possible values (R, G, B, Y)

Level 4:

M → Red

G → Green

KA → Blue

TL → Yellow

AP → 2 possible values (R, G)

KE, TN → 3 possible values (R, G, Y)

P → 4 possible values (R, G, B, Y)

Level 5:

M → Red

G → Green

KA → Blue

TL → Yellow

AP → Red

KE, TN → 2 possible values (G, Y)

P → 4 possible values (R, G, B, Y)

Level 6:

M → Red

G → Green

KA → Blue

TL → Yellow

AP → Red

KE → Yellow

TN → 1 possible value (G)

P → 4 possible value (R, G, B, Y)

Level 7:

M → Red
Gr → Green
KA → Blue
TL → Yellow
AP → Red
KE → Yellow
TN → Green (choosing 1 possible value)
P → 3 possible value (R, B, Y)

Level 8:

M → Red
Gr → Green
KA → Blue
TL → Yellow
AP → Red
KE → Yellow
TN → Green
P → Red

2nd Sub problem

SL → 4 possible values (R, Gr, B, Y)

Level 1 : SL → Green

c) Degree Heuristic

1st Subproblem:

KA:6 (Level 1) TN:4 (Level 2) M:3 (Level 3) TL:3 (Level 4) AP:3 (Level 5) KE:3 (Level 6) G:2 (Level 7) P:1 (Level 8)

Level 0 : KA, TN, M, TL, AP, KE, G, P → 4 possible values (R, G, B, Y)

Level 1 : Select KA → Red

G, M, TL, AP, TN, KE → 3 possible value (G, B, Y)

P → 4 possible values (R, G, B, Y)

Level 2 : Select TN → Green

KA → Red

KE, AP → 2 possible values (B, Y)

G, M, TL → 3 possible values (G, B, Y)

P → 3 possible values (R, B, Y)

Level 3 : Select M → Blue

KA → Red

TN → Green

G, TL → 2 possible value (G, Y)

KE, AP → 2 possible value (B, Y)

P → 3 possible value (R, B, Y)

Level 4 : Select TL → Yellow

KA → Red

TN → Green

M → Blue
AP → 1 possible value (B)
KE → 2 possible values (B, Y)
G → 2 possible values (G, Y)
P → 3 possible values (R, B, Y)

Level 5 : Select AP → Blue
KA → Red
TN → Green
M → Blue
TL → Yellow
KE → 2 possible values (B, Y)
G → 2 possible values (G, Y)
P → 3 possible values (R, B, Y)

Level 6 : Select KE → Yellow
KA → Red
TN → Green
M → Blue
TL → Yellow
AP → Blue
G → 2 possible values (G, Y)
P → 3 possible values (R, B, Y)

Level 7 : Select G → Green
KA → Red
TN → Green

M → Blue
TL → Yellow
AP → Blue
KE → Yellow
P → 3 Possible Values (R, B, Y)

Level 8: Select P → Red
KA → Red
TN → Green
M → Blue
TL → Yellow
AP → Blue
KE → Yellow
GL → Green

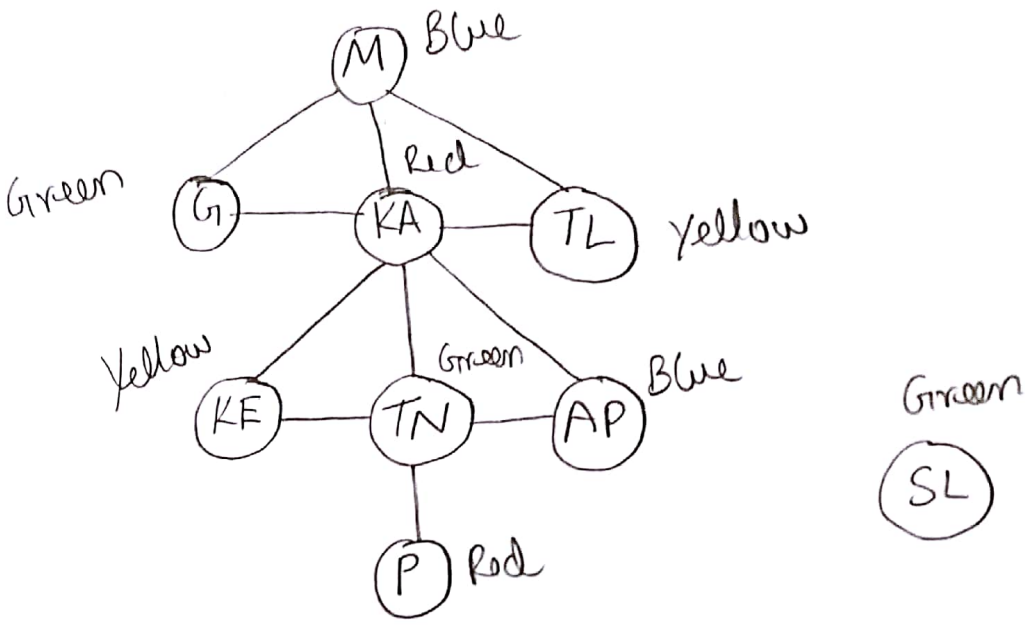
2nd Subproblem
Degree of SL ⊙
Select SL → Green

Solution:
1st Subproblem solution
KA - Red, TN - green, M - Blue, TL - yellow,
AP - Blue, KE - Yellow, GL - Green, P - red

2nd Sub problem solution:
SL - Green

Final Solution
KA - Red, TN - Green, M - Blue, TL - Yellow, AP - Blue
KE - Yellow, GL - Green, P - Red, SL - Green.

Final Solution



(2) To show that two logical statement A and B are logically equivalent

Given two knowledge bases KB1 and KB2

- In order to show that KB1 $\Leftrightarrow$ KB2, we need to first show KB1 $\Leftarrow$ KB2 as well as KB2 $\Leftarrow$ KB1
- Let us have two variable symbols and model that are globally defined.

(3)

Part a:-

In order to KB entail S_1, S_1 should be true in all worlds where KB is true.

In the given truth table, whenever KB is true S_1 is also True.

Hence, KB entails S_1.

Part b:

When we consider NOT(KB) & NOT(S_1) there are two cases where KB is True and S_1 is False (2nd and 4th Row). So, the condition for entailment doesnt get satisfied.

∴, Not(KB) does not entail NOT(S_1).

---

(4)

	A	B	C	D	
Given:	T	T	T	T	False
	T	F	T	F	False

CNF: $(\neg A \vee \neg B \vee \neg C \vee \neg D) \wedge (\neg A \vee B \vee \neg C \vee D)$

//

---

(5)

Let A: Rains on May 1, 2017

B: John gives Mary a check for $10,000 on May 2, 2017

C: Mary moves the lawn on May 3, 2017

Part a: $A \Rightarrow B$

$B \Rightarrow C$

Part b: $\overline{B}$

A

C

Part c: The contract was not violated, because when $\overline{A}$ is true $A \Rightarrow B$ holds B is true and C is also true then $B \Rightarrow C$ holds.

---

(6) (i) Shadow is a constant

dog() is a predicate

dog(shadow). There is a dog called shadow.

(ii) ~~shadow, Mary~~ John, shadow and mary are constants

gave() is a predicate

gave (shadow, Mary, John):

John gave shadow to Mary.

(iii) Shadow, Mary, Smartphone, John are constants

gave(), male() are predicates

male (shadow) => gave (smartphone, John, Mary)

If shadow is male then Mary gave smartphone to John.

(iv) Shadow, Mary, John and Laptop are constants

female(): predicate

female (shadow) => gave (Laptop, John, Mary):

If shadow is female, then Mary gave laptop to John

(v) Dog, People, John: constants

male(), gave() : predicates

gave (dogs, people, John) <=> male(dog)

If and only if it is a male dog John gives it to people

(vi) Laptop, John, Mary : constants
gave ( ) : predicate
gave (Laptop, John, Mary)

---

(7) Given first order logic knowledge base

taller (John, Bill)
$\forall x$ taller (x, Bill) $\Rightarrow$ tall (x)
Given John, Bill are constants
symbols in propositional logic:

a) taller - John - Bill = taller (John, Bill)
we have 2 constants, so x (John (or) x x | Bill

b) taller (x, Bill) = taller - John - Bill
taller - Bill - Bill

c) tall (x) = tall - John
tall - Bill

d) $(\forall x)$ taller (x, Bill) $\Rightarrow$ tall (x).

This statement is represented in propositional logic as follows,

taller - John - Bill $\Rightarrow$ tall - John.
taller - Bill - Bill $\Rightarrow$ tall - Bill.

e) A part from the above, we can also have

taller_John_John = taller (John, John)

taller_Bill_John = taller (Bill, John)

(2) Statements that should be stored in propositional logic base are

taller_John_Bill

taller_John_Bill ⇒ tall_John

taller_Bill_Bill ⇒ tall_Bill