

COMP4418  
 Assignment 1  
 Name: Ilia Chibaev  
 z3218424

Question 1:

i)  $p \wedge (q \vee r) [|= / |-] (p \wedge q) \vee (p \wedge r)$

a)  $|=$  (truth table)

			1	2	3	4	5
p	q	r	$q \vee r$	$p \wedge q$	$p \wedge r$	$p \wedge 1$	$1 \vee 2$
-----							
T	T	T	T	T	T	T	T
T	T	F	T	T	F	T	T
T	F	F	F	F	F	F	F
T	F	T	T	F	T	T	T
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	F	F	F	F	F	F
F	F	T	T	F	F	F	F

all 4  $|=$  5,  $\therefore$  Inference holds.

b)  $|-$

Negated CNF of conclusion:

$\neg((p \wedge q) \vee (p \wedge r))$

$\neg(p \wedge (q \vee r))$  | distribute

Resolution:

1.  $p \wedge (q \vee r)$  | hypothesis

2.  $\neg(p \wedge (q \vee r))$  | Negation of conclusions

3.  $[]$  | 1,2 resolution

$\therefore$  Inference holds.

ii)  $[|= / |-] p \rightarrow (q \rightarrow p)$

a)  $|=$  (truth table)

		1
p	q	$q \rightarrow p$
-----		
T	T	T
T	F	T
F	F	T
F	T	F

$\therefore$  tautology: Inference holds.

b) |-

Negated CNF of conclusion:

$\neg(p \rightarrow (q \rightarrow p))$

$\neg(\neg p \vee (\neg q \vee p))$  | Remove implications

$p \wedge \neg(\neg q \vee p)$  | DeMorgans and double negation

$p \wedge q \wedge \neg p$  | DeMorgans and double negation

Resolution:

1.  $p$  | Negation of conclusion

2.  $\neg p$  | Negation of conclusions

3. [] | 1,2 resolution

$\therefore$  Inference holds.

iii)  $\exists x. \forall y. \text{Likes}(x, y)$  [ $\models$  /  $\models$ ]  $\forall x. \exists y. \text{Likes}(x, y)$

a) KB  $\models$  CNF( $\exists x. \forall y. \text{Likes}(x, y)$ )  
 $\forall y. \text{Likes}(a, y)$  | Skolemise  
 $\text{Likes}(a, y)$  | Drop universal quantifier

alpha  $\models$  CNF( $\forall x. \exists y. \text{Likes}(x, y)$ )  
 $\text{Likes}(x, f(x))$  | Skolemise

Let I be an interpretation such that:  $I \models$  KB:

$I \models \forall y. \neg \text{Likes}(b, y)$

CNF:  $\neg \text{Likes}(b, y)$  | Drop universal quantifier

$I \models \neg \text{Likes}(b, y)$   
 $\neg \text{Likes}(b, f(b))$  |  $\{y, f(b)\}$

alpha  $\models \text{Likes}(b, f(b))$  |  $\{x, b\}$

QED: I does not entail alpha. Inference does not hold.

b) |-

CNF Hypthesis:

$\exists x. \forall y. \text{Likes}(x, y)$

$\forall y. \text{Likes}(a, y)$  | Skolemise

$\text{Likes}(a, y)$  | Drop universal quantifier

Negated CNF of conclusion:

$\neg(\forall x. \exists y. \text{Likes}(x, y))$

$\exists x. \forall y. \neg \text{Likes}(x, y)$  | Drive  $\neg$  inwards

$\forall y. \neg \text{Likes}(a, y)$  | Skolemise

$\neg \text{Likes}(a, y)$  | Drop universal quantifier

Resolution:

1.  $\text{Likes}(a, y)$  | Hypothesis

2.  $\neg \text{Likes}(a, y)$  | Negation of conclusions

3.  $\text{Likes}(a, b)$  | [1.  $\{y, b\}$ ]

4.  $\neg \text{Likes}(a, b)$  | [2.  $\{y, b\}$ ]

5. [] | 3,4 resolution

$\therefore$  Inference holds.

iv)  $\neg p \rightarrow \neg q, p \rightarrow q$  [|=/|-]  $p \leftrightarrow q$

a) |= (truth table)

		1	2	3	4
$p \quad q$		$\neg p \rightarrow \neg q$	$p \rightarrow q$	$p \leftrightarrow q$	$1 \wedge 2$
T	T	T	T	T	T
T	F	T	F	F	F
F	T	F	T	F	F
F	F	T	T	T	T

$\therefore$  all 4 |= 3: Inference holds.

b) |-

CNF Hyptheses:

1.  $\neg p \rightarrow \neg q$

$p \vee \neg q$  | Remove implication and double negation

2.  $p \rightarrow q$

$\neg p \vee q$  | Remove implication

Negated CNF of conclusion:

$\neg(p \leftrightarrow q)$

$\neg((p \rightarrow q) \wedge (q \rightarrow p))$  | Remove iff

$\neg((\neg p \vee q) \wedge (\neg q \vee p))$  | Remove implication

$\neg(\neg p \vee q) \vee \neg(\neg q \vee p)$  | DeMorgans

$(p \wedge \neg q) \vee (q \wedge \neg p)$  | DeMorgans and double negation

$(\neg p \vee \neg q) \wedge (p \vee q)$

Resolution:

1.  $p \vee \neg q$  | Hypothesis

2.  $\neg p \vee q$  | Hypothesis

3.  $\neg p \vee \neg q$  | Negation of conclusions

4.  $p \vee q$  | Negation of conclusions

5.  $\neg q$  | 1,3 resolution

6.  $\neg p$  | 2,3 resolution

7.  $p$  | 4,5 resolution

8. [] | 7,8 resolution

$\therefore$  Inference holds.

v)  $\forall x. P(x) \rightarrow Q(x), \forall x. Q(x) \rightarrow R(x), \neg R(a)$  [|=/|-]  $\neg P(a)$

a) KB |=  $\neg P(a) \vee Q(y), \neg Q(a) \vee R(y), \neg R(a)$   
 $\alpha$  |=  $\neg P(a)$

Let  $I$  be an interpretation such that:  $I \models KB$ :  
 $I \models \neg Q(y), \neg P(a) \vee Q(y)$   
 $I \models \neg P(a)$   
 $I \models \alpha$   
 QED. Inference holds.

b)  $\vdash$

CNF Hyptheses:

1.  $\forall x. P(x) \rightarrow Q(x)$   
 $\neg \forall x. P(x) \vee Q(x)$  | Remove implication  
 $\exists x. \neg P(x) \vee Q(x)$  | Drive negation inwards  
 $\exists x. \neg P(x) \vee Q(y)$  | Standardise variables  
 $\neg P(a) \vee Q(y)$  | Skolemise
2.  $\forall x. Q(x) \rightarrow R(x)$   
 $\neg Q(a) \vee R(y)$  | same steps as 1.
3.  $\neg R(a)$

Negated CNF of conclusion:

4.  $\neg(\neg P(a))$   
 $P(a)$  | Double negation

Resolution:

5.  $Q(y)$  | 1,4 resolution
6.  $Q(a)$  | [5. {y,a}]
7.  $R(y)$  | 2,6 resolution
8.  $R(a)$  | [7. {y,a}]
5. [] | 3,8 resolution

$\therefore$  Inference holds.

Question 2:

i)  $KB \models$

// From sentences

1.  $\exists x. \text{Photographer}(x)$
2.  $\exists x. \text{Photographer}(x) \rightarrow \text{SameFloor}(\text{Ivor}, x)$
3.  $\exists x. \text{MedicalStudent}(x)$
4.  $\exists x. \text{MedicalStudent}(x) \rightarrow \text{LivesImmediatelyAbove}(\text{Edwina}, x)$
5.  $\text{LivesImmediatelyAbove}(\text{Patrick}, \text{Ivor})$
6.  $\text{LawStudent}(\text{Patrick})$
7.  $\exists x. \text{AirHostess}(x)$
8.  $\exists x. \text{AirHostess}(x) \rightarrow \text{LivesOpposite}(\text{Patrick}, x)$
9.  $\exists x. \text{StoreDetective}(x)$
10.  $\exists x. \text{StoreDetective}(x) \rightarrow \text{HomeOf}(\text{Flat4}, x)$
11.  $\text{HomeOf}(\text{Flat2}, \text{Doris})$
12.  $\exists x. \text{Clerk}(x)$
13.  $\exists x. \text{HomeOf}(x, \text{Rodney})$
14.  $\exists x. \text{HomeOf}(x, \text{Rosemary})$
15.  $\exists x. \text{HomeOf}(x, \text{Ivor})$

16.  $\exists x. \text{HomeOf}(x, \text{Edwina})$   
17.  $\exists x. \text{HomeOf}(x, \text{Ivor})$   
18.  $\exists x. \text{HomeOf}(x, \text{Doris})$

// From the picture of the building

19.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat1}, x) \wedge \text{HomeOf}(\text{Flat2}, y)) \rightarrow \text{SameFloor}(x, y))$   
20.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat3}, x) \wedge \text{HomeOf}(\text{Flat4}, y)) \rightarrow \text{SameFloor}(x, y))$   
21.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat5}, x) \wedge \text{HomeOf}(\text{Flat6}, y)) \rightarrow \text{SameFloor}(x, y))$   
22.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat3}, x) \wedge \text{HomeOf}(\text{Flat1}, y)) \rightarrow \text{LivesImmediatelyAbove}(x, y))$   
23.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat4}, x) \wedge \text{HomeOf}(\text{Flat2}, y)) \rightarrow \text{LivesImmediatelyAbove}(x, y))$   
24.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat5}, x) \wedge \text{HomeOf}(\text{Flat3}, y)) \rightarrow \text{LivesImmediatelyAbove}(x, y))$   
25.  $\exists x. \exists y. ((\text{HomeOf}(\text{Flat6}, x) \wedge \text{HomeOf}(\text{Flat4}, y)) \rightarrow \text{LivesImmediatelyAbove}(x, y))$

ii) Let  $I$  be an interpretation such that:  $I \models \text{KB}$   
Case 1:  $I \models \text{HomeOf}(\text{Flat4}, \text{Ivor})$

iii) No, need to make a couple of assumptions:  
19.  $\text{AirHostess}(x) \rightarrow \text{Female}(x)$   
20.  $\text{Female}(\text{Rosemary})$   
21.  $\neg \text{Female}(\text{Rodney})$

Flat 5: Rosemary, Air Hostess,  
Flat 6: Patrick, Law Student,  
Flat 3: Edwina, Photographer,  
Flat 4: Ivor, Store Detective,  
Flat 1: Rodney, Medical Student,  
Flat 2: Doris, Clerk

Question 3 report:

The solution is implemented in C++17. To build the executable, run `make`. To execute, type `./assnq1 'your query'`

The solution contains 2 classes, and 2 namespaces (for static functions) in separate files (each with `.cpp` and `.h`).

The `Sequent` class is used to parse, store, and represent (print) a sequent in the form eg. `'[x, y] seq [y, x neg z]'`.

The `SearchTree` files contain a struct for a `Node`, with a possible pair of sequent values (to represent backward split into 2 sequents), a list of pointers to children nodes, and a pointer to it's parent node. The `SearchTree` class initialises from and stores a root node, and contains functions to build the search tree from an input sequent.

The Rule namespace contains functions to check whether a formula can be transformed according to a rule, and to perform this transformation.

The Utils namespace contains functions for string processing using regex functions. Some examples are finding the outermost rule keyword (outside all brackets), stripping brackets, and splitting a string into a list.

#### Question 4:

##### Frames

(i) how the method represents knowledge;

This method represents knowledge in an object oriented style. A frame represents an object, with named attributes and which has relations with other frames. The frame class follows a hierarchical taxonomy, an example of which could be: Mammal({attributes}) -> Primate({attributes}) -> Man({attributes}) -> Steve({attributes}). Frame attributes (called slots) can also be references to other frames.

There are individual frames - which represent instances of objects, and generic frames - which represent categories or classes of objects. Individual frames have a field indicating their class which refers to a generic frame. Frames have inheritance of attributes from their parent classes, where the values of these attributes are not defined on instantiation.

Generic frames can also have attributes which are filled with attached procedures. Any constraints on attribute values are expressed through these stored procedures. Procedures are expressed as attribute values under tags: IF-ADDED and IF-NEEDED, which represent whether the attribute is being newly created, or is being queried.

(ii) describes how inference works for reasoning with that knowledge representation.

Inference works by asking by querying the values of attributes in a frame, and this query cascading through the knowledge-base until a value is reached and returned. Since frame attributes can contain stored procedures, this cascade can alter the knowledge base, create new frames, and trigger other procedure calls.

If the query does not produce a result then the value of the slot is considered unknown.