COMP4418
Assignment 1
Name: Ilia Chibaev
z3218424

Question 1:

i) p ∧ (q ∨ r)[|=/|-](p ∧ q) ∨ (p ∧ r)

a) |= (truth table)

	1	2	3	4	5
p q r	q ∨ r	p ∧ q	p ∧ r	p ∧ 1	1 ∨ 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, :. Inference holds.

b) |-
Negated CNF of conclusion:
¬((p ∧ q) ∨ (p ∧ r))
¬(p ∧ (q ∨ r)) | distribute

Resolution:
1. p ∧ (q ∨ r) | hypothesis
2. ¬(p ∧ (q ∨ r)) | Negation of conclusions
3. [] | 1,2 resolution

:. Inference holds.

ii) [|=/|-] p → (q → p)

a) |= (truth table)

	1	
p q	q → p	p → 1
T T	T	T
T F	T	T
F F	T	T
F T	F	T

:. tautology: Inference holds.

```
    b) |-
        Negated CNF of conclusion:
        ¬(p → (q → p))
        ¬(¬p v (¬q v p))  | Remove implications
        p ∧ ¬(¬q v p)     | DeMorgans and double negation
        p ∧ q ∧ ¬p        | DeMorgans and double negation

        Resolution:
        1. p              | Negation of conclusion
        2. ¬p             | Negation of conclusions
        3. []             | 1,2 resolution

        :. Inference holds.


iii) ∃x.∀y.Likes(x,y) [|=/|-] ∀x.∃y.Likes(x,y)

    a)  KB |=    CNF(∃x.∀y.Likes(x,y))
                 ∀y.Likes(a,y)          | Skolomise
                 Likes(a,y)             | Drop universal quantifier

        alpha |= CNF(∀x.∃y.Likes(x,y))
                 Likes(x,f(x))          | Skolomise

        Let I be an interpretation such that: I |= KB:
        I |= ∀y.¬Likes(b,y)
        CNF:  ¬Likes(b,y)               | Drop universal quantifier

        I |=  ¬Likes(b,y)
              ¬Likes(b,f(b))            | {y, f(b)}

        alpha |=  Likes(b, f(b))        | {x, b}

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

    b) |-
        CNF Hypthesis:
        ∃x.∀y.Likes(x,y)
        ∀y.Likes(a,y)       | Skolomise
        Likes(a,y)          | Drop universal quantifier

        Negated CNF of conclusion:
        ¬(∀x.∃y.Likes(x,y))
        ∃x.∀y.¬Likes(x,y)   | Drive ¬ inwards
        ∀y.¬Likes(a,y)      | Skolomise
        ¬Likes(a,y)         | Drop universal quantifier

        Resolution:
        1. Likes(a,y)       | Hypothesis
        2. ¬Likes(a,y)      | Negation of conclusions
        3. Likes(a,b)       | [1. {y,b}]
        4. ¬Likes(a,b)      | [2. {y,b}]
```

```
    5. []                | 3,4 resolution

    :. Inference holds.
```

iv) ¬p → ¬q, p → q [|=/|-] p ↔ q

a) |= (truth table)

p	q	1: ¬p → ¬q	2: p → q	3: p ↔ q	4: 1 ∧ 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

:. all 4 |= 3: Inference holds.

b) |-

```
CNF Hyptheses:
1. ¬p → ¬q
   p ∨ ¬q          | Remove implication and double negation

2. p → q
   ¬p ∨ q          | Remove implication

Negated CNF of conclusion:
¬(p ↔ q)
¬((p → q) ∧ (q → p))        | Remove iff
¬((¬p ∨ q) ∧ (¬q ∨ p))      | Remove implication
¬(¬p ∨ q) ∨ ¬(¬q ∨ p))      | DeMorgans
(p ∧ ¬q) ∨ (q ∧ ¬p)         | DeMorgans and double negation
((p ∧ ¬q) ∨ ¬p) ∧ ((p ∧ ¬q) ∨ q) | and over or
((p ∨ ¬p) ∧ (¬q ∨ ¬p)) ∧ ((p ∨ q) ∧ (¬q ∨ q)) |and over or
(¬p ∨ ¬q) ∧ (p ∨ q)         | resolve terms

Resolution:
1. p ∨ ¬q                   | Hypothesis
2. ¬p ∨ q                   | Hypothesis
3. ¬p ∨ ¬q                  | Negation of conclusions
4. p ∨ q                    | Negation of conclusions
5. ¬q                       | 1,3 resolution
6. ¬p                       | 2,3 resolution
7. p                        | 4,5 resolution
8. []                       | 7,8 resolution

:. Inference holds.
```

v) ∀x.P(x) → Q(x), ∀x.Q(x) → R(x), ¬R(a) [|=/|-] ¬P(a)

a) KB |= ¬P(a) ∨ Q(y), ¬Q(a) ∨ R(y), ¬R(a)

```
        alpha |= ¬P(a)

        Let I be an interpretation such that: I |= KB:
        I |= ¬Q(y), ¬P(a) ∨ Q(y)
        I |= ¬P(a)
        I |= alpha
        QED. Inference holds.

    b) |-
        CNF Hyptheses:
        1.  ∀x.P(x) → Q(x)
            ¬∀x.P(x) ∨ Q(x)      | Remove implication
            ∃x.¬P(x) ∨ Q(x)      | Drive negation inwards
            ∃x.¬P(x) ∨ Q(y)      | Standardise variables
            ¬P(a) ∨ Q(y)         | Skolomise

        2. ∀x.Q(x) → R(x)
           ¬Q(a) ∨ R(y)          | same steps as 1.

        3. ¬R(a)

        Negated CNF of conclusion:
        4.  ¬(¬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

        :. Inference holds.


Question 2:

i) KB |=
// From sentences
1.  ∃x.Photographer(x)
2.  ∃x.Photographer(x) → SameFloor(Ivor, x)
3.  ∃x.MedicalStudent(x)
4.  ∃x.MedicalStudent(x) → LivesImmediatelyAbove(Edwina, x)
5.  LivesImmediatelyAbove(Patrick, Ivor)
6.  LawStudent(Patrick)
7.  ∃x.AirHostess(x)
8.  ∃x.AirHostess(x) → LivesOpposite(Patrick, x)
9.  ∃x.StoreDetective(x)
10. ∃x.StoreDetective(x) → HomeOf(Flat4, x)
11. HomeOf(Flat2, Doris)
12. ∃x.Clerk(x)
13. ∃x.HomeOf(x, Rodney)
```

```
14. ∃x.HomeOf(x, Rosemary)
15. ∃x.HomeOf(x, Ivor)
16. ∃x.HomeOf(x, Edwina)
17. ∃x.HomeOf(x, Ivor)
18. ∃x.HomeOf(x, Doris)

// From the picture of the building
19. ∃x.∃y((HomeOf(Flat1, x) ∧ HomeOf(Flat2, y)) → SameFloor(x, y))
20. ∃x.∃y((HomeOf(Flat3, x) ∧ HomeOf(Flat4, y)) → SameFloor(x, y))
21. ∃x.∃y((HomeOf(Flat5, x) ∧ HomeOf(Flat6, y)) → SameFloor(x, y))
22. ∃x.∃y((HomeOf(Flat3, x) ∧ HomeOf(Flat1, y)) →
     LivesImmediatelyAbove(x, y))
23. ∃x.∃y((HomeOf(Flat4, x) ∧ HomeOf(Flat2, y)) →
     LivesImmediatelyAbove(x, y))
24. ∃x.∃y((HomeOf(Flat5, x) ∧ HomeOf(Flat3, y)) →
     LivesImmediatelyAbove(x, y))
25. ∃x.∃y((HomeOf(Flat6, x) ∧ HomeOf(Flat4, y)) →
     LivesImmediatelyAbove(x, y))

ii) Let I be an interpretation such that: I |= KB
    Case 1: I |= HomeOf(Flat4, Ivor)


iii) No, need to make a couple of assumptions:
    19. AirHostess(x) → Female(x)
    20. Female(Rosemary)
    21. ¬Female(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 sequenet 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 tranformed according to a rule, and to perform this transformation.

The Utils namespaces 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 represent 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 heirarchical 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 refer 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 then 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.