COMP4418, 2017–Assignment 1

1. [20 Marks] (Propositional Inferences)

Prove whether or not the following inferences hold in propositional logic using the truth table method.

(a) $p \vee (q \wedge r) \models (p \vee q) \wedge (p \vee r)$ **valid**

p	q	r	$q \wedge r$	$p \vee q$	$p \vee r$	$p \vee (q \wedge r)$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
F	T	T	T	T	T	T	T
T	F	F	F	T	T	T	T
F	T	F	F	T	F	F	F
F	F	T	F	F	T	F	F
F	F	F	F	F	F	F	F

In all rows where $p \vee (q \wedge r)$ is true, $(p \vee q) \wedge (p \vee r)$ is also true. Therefore, inference is valid.

(b) $\models p \rightarrow (q \rightarrow p)$ **valid**

p	q	$q \rightarrow p$	$p \rightarrow (q \rightarrow p)$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

Last column is always true no matter what truth assignment to the atoms p and q. Therefore $p \rightarrow (q \rightarrow p)$ is a tautology.

(c) $p \rightarrow q \models \neg p \rightarrow \neg q$ **NOT valid**

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg p \rightarrow \neg q$
T	T	F	F	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

In the fourth row where $p \rightarrow q$ is true, $\neg p \rightarrow \neg q$ is false. Therefore, inference is not valid.

(d) $p \rightarrow q, \neg p \rightarrow \neg q \models \neg p \leftrightarrow \neg q$ **valid**

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg p \rightarrow \neg q$	$\neg p \leftrightarrow \neg q$
T	T	F	F	T	T	T
T	F	F	T	F	T	F
F	T	T	F	T	F	F
F	F	T	T	T	T	T

In all rows where both $p \rightarrow q$ and $\neg p \rightarrow \neg q$ are true, $\neg p \leftrightarrow \neg q$ is also true. Therefore, inference is valid.

(e) $\neg q \rightarrow \neg p, \neg r \rightarrow \neg q \models p \rightarrow r$ **valid**

q	p	r	$\neg q$	$\neg p$	$\neg r$	$\neg q \rightarrow \neg p$	$\neg r \rightarrow \neg q$	$p \rightarrow r$

T	T	T	F	F	F	T	T	T
T	T	F	F	F	T	T	F	F
T	F	T	F	T	F	T	T	T
F	T	T	T	F	F	F	T	T
T	F	F	F	T	T	T	F	T
F	T	F	T	F	T	F	T	F
F	F	T	T	T	F	T	T	T
F	F	F	T	T	T	T	T	T

In all rows where both $\neg q \rightarrow \neg p$ and $\neg r \rightarrow \neg q$ are true, $p \rightarrow r$ is also true. Therefore, inference is valid.

Prove whether or not the following inferences hold in propositional logic using resolution.

(f) $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$ **valid**

$\mathrm{CNF}\big(p \wedge (q \vee r)\big) \equiv p \wedge (q \vee r)$

$\mathrm{CNF}(\neg\big((p \wedge q) \vee (p \wedge r)\big)) \equiv (\neg p \vee \neg q) \wedge (\neg p \vee \neg r)$

Proof:

1. p (Hypothesis)

2. $q \vee r$ (Hypothesis)

3. $\neg p \vee \neg q$ (Negation of Conclusion)

4. $\neg p \vee \neg r$ (Negation of Conclusion)

5. $\neg p \vee r$ (2, 3 Resloution)

6. $\neg p$ (4, 5 Resloution)

7. ■ (1, 6 Resloution)

(g) $p \vdash p \rightarrow q$ **NOT valid**

$\mathrm{CNF}(p) \equiv p$

$\mathrm{CNF}(\neg(p \rightarrow q)) \equiv p \wedge \neg q$

1. p (Hypothesis)

2. $\neg q$ (Negation of Conclusion)

(h) $p \leftrightarrow q \vdash (q \leftrightarrow r) \rightarrow (p \leftrightarrow r)$ **valid**

$\mathrm{CNF}(p \leftrightarrow q) \equiv (p \rightarrow q) \wedge (q \rightarrow p) \equiv (\neg p \vee q) \wedge (\neg q \vee p)$

$\mathrm{CNF}(\neg\big((q \leftrightarrow r) \rightarrow (p \leftrightarrow r)\big)) \equiv (q \vee \neg r) \wedge (r \vee \neg q) \wedge (p \vee r) \wedge (\neg p \vee \neg r)$

1. $\neg p \vee q$ (Hypothesis)

2. $\neg q \vee p$ (Hypothesis)

3. $q \vee \neg r$ (Negation of Conclusion)

4. $r \vee \neg q$ (Negation of Conclusion)

5. $p \vee r$ (Negation of Conclusion)

6. $\neg p \vee \neg r$ (Negation of Conclusion)

7. $\neg r \vee p$ (2, 3 Resloution)

8. $q \vee \neg r$ (1, 7 Resloution)

9. ■ (4, 8 Resloution)

(i) $\neg p \wedge \neg q \vdash p \leftrightarrow q$ **valid**

$\mathrm{CNF}(\neg p \wedge \neg q) \equiv \neg p \wedge \neg q$

$CNF(\neg(p \leftrightarrow q)) \equiv (p \lor q) \land (\neg p \lor \neg q)$

1. $\neg p$ (Hypothesis)

2. $\neg q$ (Hypothesis)

3. $p \lor q$ (Negation of Conclusion)

4. $\neg p \lor \neg q$ (Negation of Conclusion)

5. q (1, 3 Resloution)

6. ■ (2, 5 Resloution)

(j) $\neg q \rightarrow \neg p, \neg r \rightarrow \neg q \vdash p \rightarrow r$ **valid**

$CNF(\neg q \rightarrow \neg p) \equiv q \lor \neg p$

$CNF(\neg r \rightarrow \neg q) \equiv r \lor \neg q$

$CNF(\neg(p \rightarrow r)) \equiv p \land \neg r$

1. $q \lor \neg p$ (Hypothesis)

2. $r \lor \neg q$ (Hypothesis)

3. p (Negation of Conclusion)

4. $\neg r$ (Negation of Conclusion)

5. q (1, 3 Resloution)

6. r (2, 5 Resloution)

7. ■ (4, 6 Resloution)

2. [30 Marks] (Logic Puzzle)

Daisy and Donald Duck took their nephews aged 4, 5 and 6 on an outing. Each boy wore a tee-shirt with a different design on it and of a different colour.

(a)

1. Huey is younger than the boy in the green tee-shirt

$\forall x \forall a \forall b. \left(\left(\mathrm{colour}(x, \mathrm{green}) \wedge \mathrm{age}(\mathrm{huey}, a) \wedge \mathrm{age}(x, b) \right) \rightarrow a < b \right)$

2. The five year-old wore the tee-shirt with the camel design

$\forall x. (\mathrm{age}(x, 5) \rightarrow \mathrm{design}(x, \mathrm{camel}))$

3. Dewey's tee-shirt was yellow

$\mathrm{colour}(\mathrm{dewey}, \mathrm{yellow})$

4. Louie's tee-shirt bore the giraffe design

$\mathrm{design}(\mathrm{louie}, \mathrm{girrafe})$

5. The panda design was not featured on the white tee-shirt

$\neg \exists x. (\mathrm{design}(x, \mathrm{panda}) \wedge \mathrm{color}(x, \mathrm{white}))$

(b)

Using my formalisation in part (2a), it is possible to conclude the age of each boy together with the colour and design of the tee-shirt they're wearing.

name	age	colour	design
Huey	5	white	camel
Dewy	4	yellow	panda
Lowie	6	green	giraffe

Prove semantically:

$$S = \{\forall a \forall b \forall c. (colour(huey, a) \wedge \mathrm{colour}(\mathrm{dewey}, b) \wedge \mathrm{colour}(\mathrm{louie}, c) \rightarrow a \neq b \neq c), \forall a \forall b \forall c. (age(huey, a) \wedge \mathrm{age}(\mathrm{dewey}, b) \wedge \mathrm{age}(\mathrm{louie}, c) \rightarrow a \neq b \neq c), \forall a \forall b \forall c. (design(huey, a) \wedge \mathrm{design}(\mathrm{dewey}, b) \wedge \mathrm{design}(\mathrm{louie}, c) \rightarrow a \neq b \neq c), \forall x \forall a \forall b. \left(\left(\mathrm{colour}(x, \mathrm{green}) \wedge \mathrm{age}(\mathrm{huey}, a) \wedge \mathrm{age}(x, b) \right) \rightarrow a < b, \forall x. (\mathrm{age}(x, 5) \rightarrow \mathrm{design}(x, \mathrm{camel})), \mathrm{colour}(\mathrm{dewey}, \mathrm{yellow}), \mathrm{design}(\mathrm{louie}, \mathrm{girrafe}), \neg \exists x. (\mathrm{design}(x, \mathrm{panda}) \wedge \mathrm{color}(x, \mathrm{white}))\}$$

all that is required.

$$\alpha = age(huey, 5) \wedge colour(huey, white) \wedge design(huey, camel) \wedge age(dewy, 4) \wedge colour(dewy, yellow) \wedge design(dewy, panda) \wedge age(lowie, 6) \wedge colour(lowie, green) \wedge design(lowie, giraffe)$$

Claim: $S| = \alpha$

Proof:

Let I be any interpretation such that $I| = S$.

Case 1: I |= $colour(huey, green)$

Because statement 1, Huey is younger than the boy in the green tee-shirt. We can conclude that Huey is not in green tee-shirt.

Case 2: $I \models colour(huey, white)$

Because statement 4 and 5, the design of Huey is not giraffe or panda.

$\therefore I \models design(huey, camel)$

Because statement 2, the boy with camel design is five years old.

$\therefore I \models age(huey, 5)$

$\therefore I \models age(huey, 5) \land colour(huey, white) \land design(huey, camel)$

Because statement 3 and above statements of Huey, the colour of Lowie is not white and yellow.

$\therefore I \models colour(lowie, green)$

Because statement 1 and above statements of Huey, Lowie is older than 5.

$\therefore I \models age(lowie, 6)$

$\therefore I \models age(huey, 5) \land colour(huey, white) \land design(huey, camel) \land age(lowie, 6) \land colour(lowie, green) \land design(lowie, giraffe)$

$\therefore I \models age(dewy, 4) \land colour(dewy, yellow) \land design(dewy, panda)$

$\therefore I \models age(huey, 5) \land colour(huey, white) \land design(huey, camel) \land age(dewy, 4) \land colour(dewy, yellow) \land design(dewy, panda) \land age(lowie, 6) \land colour(lowie, green) \land design(lowie, giraffe)$

$\therefore S \models \alpha$

Case 3: $I \models colour(huey, yellow)$

Because statement 3, Dewey's tee-shirt was yellow. We can conclude that Huey is not in yellow tee-shirt.

3. [50 Marks] (Automated Theorem Proving)

I am using **Python 3.5** to deal with this problem.

All the outputs are here. And I also write some annotations in my code in order to understand it much easier.

The main procedure is:

First: read the input sequent and separate it in to two lists—left and right. For each list, either left list or right list, different formulae are in different and individual list.

For example, [[(neg p) or p], [r]] includes two formulae: (neg p) or p and r.

Second: define a function to classify the priority of operators, in order to deal with the sequence in correct order.

For example: we try to set 'imp' and 'iff' a value of 3, 'and' and 'or' a value of 2, 'neg' a value of 1. In addition, all the items in brackets (including brackets) have the value of 0.

Third: define all the rules mentioned in the assignments.

For Rule 1, I decided to decompose all of the formulae into single atomic form, in order to make the procedure more clearly.

Forth: in a large loop, we need to deal with left lists and right lists, in every loop, first left and then right.

Every time we only deal with the formulae with the highest priority, and after that, we should compute the priority again. And if all the priority is 0, we need to delete the brackets and compute all the priority again.

We should record each new result every time and the rules we used.

Every time we should use the new result to replace the old one. When we meet a formula separate into two formulae, add the new one into our original lists—which include original left and right list. After that, the big loop will be changed to deal with the original list and the new list.

If we find all the list in this loop can become an atomic formula, break. Otherwise, if the loop time bigger than 100, break and output 'False.'.

Fifth: Output all the results, in the reverse order. (The most difficult part for me……It is so hard to match the rules and result lists in right order. Anyway, I made it!)

```
H:\Python 3.5.3\projects>assn1q3.py '[] seq [neg p or p]'
True.
---------
Proofs:
1.   [ p ] seq [ p ]         Rule 1
2.   [ ] seq [ neg p, p ]    Rule P2a
3.   [ ] seq [ neg p or p ]  Rule P4a
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[neg(p or q)] seq [neg p]'
True.
---------
Proofs:
1.  [ p ] seq [ p, q ]                   Rule 1
2.  [ p ] seq [ ( p or q ) ]             Rule P4a
3.  [ ] seq [ neg p, ( p or q ) ]        Rule P2a
4.  [ neg ( p or q ) ] seq [ neg p ]     Rule P2b
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p] seq [q imp p]'
True.
---------
Proofs:
1.  [ p, q ] seq [ p ]      Rule 1
2.  [ p ] seq [ q imp p ]   Rule P5a
-----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p] seq [p or q]'
True.
---------
Proofs:
1.  [ p ] seq [ p, q ]     Rule 1
2.  [ p ] seq [ p or q ]   Rule P4a
-----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[(p and q) and r] seq [p and (q and r)]'
True.
---------
Proofs:
1.  [ p, q, r ] seq [ r ]                                 Rule 1
2.  [ p and q, r ] seq [ r ]                              Rule P3b
3.  [ p, q, r ] seq [ q ]                                 Rule 1
4.  [ p and q, r ] seq [ q ]                              Rule P3b
5.  [ p and q, r ] seq [ q and r ]                        Rule P3a
6.  [ p, q, r ] seq [ p ]                                 Rule 1
7.  [ p and q, r ] seq [ p ]                              Rule P3b
8.  [ ( p and q ), r ] seq [ p and ( q and r ) ]          Rule P3a
9.  [ ( p and q ) and r ] seq [ p and ( q and r ) ]       Rule P3b
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p iff q] seq [neg (p iff neg q)]'
True.
---------
Proofs:
1.  [ q ] seq [ p, q, p ]                       Rule 1
2.  [ ] seq [ p, q, p, neg q ]                  Rule P2a
3.  [ p, q, q ] seq [ p ]                       Rule 1
4.  [ p, q ] seq [ p, neg q ]                   Rule P2a
5.  [ p ] seq [ p, q, q ]                       Rule 1
6.  [ p, neg q ] seq [ p, q ]                   Rule P2b
7.  [ ( p iff neg q ) ] seq [ p, q ]            Rule P6b
8.  [ ] seq [ neg ( p iff neg q ), p, q ]       Rule P2a
9.  [ p, q, p ] seq [ q ]                       Rule 1
10. [ p, q, p, neg q ] seq [ ]                  Rule P2b
11. [ p, q, ( p iff neg q ) ] seq [ ]           Rule P6b
12. [ p, q ] seq [ neg ( p iff neg q ) ]        Rule P2a
13. [ p iff q ] seq [ neg ( p iff neg q ) ]     Rule P6b
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p iff q] seq [(q iff r)imp(p iff r)]'
True.
---------
Proofs:
1.  [ p ] seq [ r, p, q, q, r ]                         Rule 1
2.  [ p, q, p ] seq [ r, q, r ]                         Rule 1
3.  [ q, r, p ] seq [ r, p, q ]                         Rule 1
4.  [ r ] seq [ p, p, q, q, r ]                         Rule 1
5.  [ ] seq [ p iff r, p, q, q, r ]                     Rule P6a
6.  [ p, q, q, r, p ] seq [ r ]                         Rule 1
7.  [ p, q, r ] seq [ p, q, r ]                         Rule 1
8.  [ p, q ] seq [ p iff r, q, r ]                      Rule P6a
9.  [ q, r, r ] seq [ p, p, q ]                         Rule 1
10. [ q, r ] seq [ ( p iff r ), p, q ]                  Rule P6a
11. [ ( q iff r ) ] seq [ ( p iff r ), p, q ]           Rule P6b
12. [ ] seq [ ( q iff r ) imp ( p iff r ), p, q ]       Rule P5a
13. [ p, q, q, r, r ] seq [ p ]                         Rule 1
14. [ p, q, q, r ] seq [ ( p iff r ) ]                  Rule P6a
15. [ p, q, ( q iff r ) ] seq [ ( p iff r ) ]           Rule P6b
16. [ p, q ] seq [ ( q iff r ) imp ( p iff r ) ]        Rule P5a
17. [ p iff q ] seq [ ( q iff r ) imp ( p iff r ) ]     Rule P6b
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[] seq [((neg p)and(neg q))imp(p iff q)]'
True.
---------
Proofs:
1.  [ p ] seq [ q, p, q ]                                   Rule 1
2.  [ neg q, p ] seq [ q, p ]                               Rule P2b
3.  [ neg p, neg q, p ] seq [ q ]                           Rule P2b
4.  [ q ] seq [ p, p, q ]                                   Rule 1
5.  [ neg q, q ] seq [ p, p ]                               Rule P2b
6.  [ ( neg p ), ( neg q ), q ] seq [ p ]                   Rule P2b
7.  [ ( neg p ), ( neg q ) ] seq [ ( p iff q ) ]            Rule P6a
8.  [ ( ( neg p ) and ( neg q ) ) ] seq [ ( p iff q ) ]     Rule P3b
9.  [ ] seq [ ( ( neg p ) and ( neg q ) ) imp ( p iff q ) ] Rule P5a
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p iff q] seq [(p and q)or((neg p)and(neg q))]'
True.
---------
Proofs:
1.  [ q ] seq [ q, p, q ]                                       Rule 1
2.  [ ] seq [ q, neg q, p, q ]                                  Rule P2a
3.  [ p, q, q ] seq [ q ]                                       Rule 1
4.  [ p, q ] seq [ q, neg q ]                                   Rule P2a
5.  [ q ] seq [ p, p, q ]                                       Rule 1
6.  [ ] seq [ p, neg q, p, q ]                                  Rule P2a
7.  [ p ] seq [ q, p, q ]                                       Rule 1
8.  [ ] seq [ q, ( neg p ), p, q ]                              Rule P2a
9.  [ ] seq [ q, ( neg p ) and ( neg q ), p, q ]                Rule P3a
10. [ p, q, q ] seq [ p ]                                       Rule 1
11. [ p, q ] seq [ p, neg q ]                                   Rule P2a
12. [ p, q, p ] seq [ q ]                                       Rule 1
13. [ p, q ] seq [ q, ( neg p ) ]                               Rule P2a
14. [ p, q ] seq [ q, ( neg p ) and ( neg q ) ]                 Rule P3a
15. [ p ] seq [ p, p, q ]                                       Rule 1
16. [ ] seq [ p, ( neg p ), p, q ]                              Rule P2a
17. [ ] seq [ p, ( neg p ) and ( neg q ), p, q ]                Rule P3a
18. [ ] seq [ ( p and q ), ( ( neg p ) and ( neg q ) ), p, q ]  Rule P3a
19. [ ] seq [ ( p and q ) or ( ( neg p ) and ( neg q ) ), p, q ] Rule P4a
20. [ p, q, p ] seq [ p ]                                       Rule 1
21. [ p, q ] seq [ p, ( neg p ) ]                               Rule P2a
22. [ p, q ] seq [ p, ( neg p ) and ( neg q ) ]                 Rule P3a
23. [ p, q ] seq [ ( p and q ), ( ( neg p ) and ( neg q ) ) ]   Rule P3a
24. [ p, q ] seq [ ( p and q ) or ( ( neg p ) and ( neg q ) ) ] Rule P4a
25. [ p iff q ] seq [ ( p and q ) or ( ( neg p ) and ( neg q ) ) ] Rule P6b
----------
QED.
```

```
H:\Python 3.5.3\projects>assn1q3.py '[p imp q,(neg r)imp(neg q)] seq [p imp r]'
True.
---------
Proofs:
1.  [ p, r ] seq [ r, p ]                                 Rule 1
2.  [ p ] seq [ r, p, ( neg r ) ]                         Rule P2a
3.  [ ] seq [ p imp r, p, ( neg r ) ]                     Rule P5a
4.  [ q, p, r ] seq [ r ]                                 Rule 1
5.  [ q, p ] seq [ r, neg r ]                             Rule P2a
6.  [ p ] seq [ r, p, q ]                                 Rule 1
7.  [ ( neg q ), p ] seq [ r, p ]                         Rule P2b
8.  [ ( neg q ) ] seq [ p imp r, p ]                      Rule P5a
9.  [ ( neg r ) imp ( neg q ) ] seq [ p imp r, p ]        Rule P5b
10. [ q, p ] seq [ r, q ]                                 Rule 1
11. [ q, ( neg q ), p ] seq [ r ]                         Rule P2b
12. [ q, ( neg r ) imp ( neg q ), p ] seq [ r ]           Rule P5b
13. [ q, ( neg r ) imp ( neg q ) ] seq [ p imp r ]        Rule P5a
14. [ p imp q , ( neg r ) imp ( neg q ) ] seq [ p imp r ]  Rule P5b
----------
QED.
```