

# COMP4418, 2019–Assignment 1

Due: 23:59:59pm Sunday 13 October (End of Week 4)

Late penalty: 10 marks per day.

Worth: 15%.

---

This assignment consists of four questions. The first two questions and the fourth question require written answers only. The third question requires some programming and a written report.

1. [20 Marks] (Logical Inference)

For each of the following inferences:

- (a) prove whether or not the following inferences hold using a suitable semantic method ( $\models$ ); and,
- (b) prove whether or not the following inferences hold syntactically using resolution ( $\vdash$ ).

In each case you must provide a proof and clearly state whether the inference holds or whether the inference does not hold.

- (i)  $p \wedge (q \vee r) \models / \vdash (p \wedge q) \vee (p \wedge r)$
- (ii)  $\models / \vdash p \rightarrow (q \rightarrow p)$
- (iii)  $\exists x.\forall y.Likes(x, y) \models / \vdash \forall x.\exists y.Likes(x, y)$
- (iv)  $\neg p \rightarrow \neg q, p \rightarrow q \models / \vdash p \leftrightarrow q$
- (v)  $\forall x.P(x) \rightarrow Q(x), \forall x.Q(x) \rightarrow R(x), \neg R(a) \models / \vdash \neg P(a)$

2. [20 Marks] (Logic Puzzle)

Flatmates, from Logic Problems, Issue 10, page 35. The following logic puzzle appears in the lecture notes and is solved using Prolog. Here we will determine a solution and argue using interpretations as explained in lectures. Six people live in a three-storey block of studio flats laid out as in the plan.

Flat 5	Flat 6
Flat 3	Flat 4
Flat 1	Flat 2

- Ivor and the photographer live on the same floor.
- Edwina lives immediately above the medical student.
- Patrick, who is studying law, lives immediately above Ivor, and opposite the air hostess.
- Flat4 is the home of the store detective.
- Doris lives in Flat2.
- Rodney and Rosemary are 2 of the residents in the block of flats.
- One of the residents is a clerk.

- (i) Represent these facts as sentences in first-order logic.
- (ii) From the clues given, work out the name and situation of the resident of each flat. Using your formalisation in part (2a), is it possible to determine the name and situation of the resident of each flat? Show semantically how you determined your answer.
- (iii) If your answer to part (2b) was ‘no’, indicate what further sentences you would need to add to your formalisation so that name and situation of the resident of each flat.

3. [50 Marks] (Automated Theorem Proving)

In 1958 the logician Hao Wang implemented one of the first automated theorem provers. He succeeded in writing several programs capable of automatically proving a majority of theorems from the first five chapters of Whitehead and Russell’s *Principia Mathematica* (in fact, his program managed to prove over 200 of these theorems “within about 37 minutes, and 12/13 of the time is used for read-in and print-out”). This was an impressive achievement at the time; previous attempts had only succeeded in proving a handful of the theorems in *Principia Mathematica*.

# Background

Wang's idea is based around the notion of a *sequent* (this idea had been introduced years earlier by Gentzen) and the manipulation of sequents. A sequent is essentially a list of formulae on either side of a sequent (or provability) symbol  $\vdash$ . The sequent  $\pi \vdash \rho$ , where  $\pi$  and  $\rho$  are strings (i.e., lists) of formulae, can be read as “the formulae in the string  $\rho$  follow from the formulae in the string  $\pi$ ” (or, equivalently, “the formulae in string  $\pi$  prove the formulae in string  $\rho$ ”).

To prove whether a given sequent is true all you need to do is start from some basic sequents and successively apply a series of rules that transform sequents until you end up with the sequent you desire. This process is detailed below.

Additionally, determining whether a formula  $\phi$  is a theorem, is equivalent to determining whether the sequent  $\emptyset \vdash \phi$  is true (e.g.,  $\vdash \neg\phi \vee \phi$ ).

## Formulae

### Connectives

We allow the following connectives in decreasing order of precedence:

$\neg$  — negation

$\wedge$  — conjunction;  $\vee$  — disjunction (both same precedence)

$\rightarrow$  — implication;  $\leftrightarrow$  — biconditional (both same precedence).

### Formula

- A propositional symbol (e.g.,  $p$ ,  $q$ , ...) is an *atomic* formula (and thus a formula).
- If  $\phi$ ,  $\psi$  are formulae, then  $\neg\phi$ ,  $\phi \wedge \psi$ ,  $\phi \vee \psi$ ,  $\phi \rightarrow \psi$ ,  $\phi \leftrightarrow \psi$  are formulae.

### Sequent

If  $\pi$  and  $\rho$  are strings of formulae (possibly empty strings) and  $\phi$  is a formula, then  $\pi$ ,  $\phi$ ,  $\rho$  is a string and  $\pi \vdash \rho$  is a sequent.

## Rules

The logic consists of the following sequent rules. The first rule (P1) gives a characterisation of simple theorems. The remaining rules are simply ways of transforming sequents into new sequents. The manner in which you can construct a proof for a sequent to determine whether it holds or not is given below.

**P1** Initial Rule: If  $\lambda$ ,  $\zeta$  are strings of atomic formulae, then  $\lambda \vdash \zeta$  is a theorem if some atomic formula occurs on both side of the sequent  $\vdash$ .

In the following ten rules  $\lambda$  and  $\zeta$  are always strings (possibly empty) of formulae.

**P2a** Rule  $\vdash \neg$ : If  $\phi$ ,  $\zeta \vdash \lambda$ ,  $\rho$ , then  $\zeta \vdash \lambda$ ,  $\neg\phi$ ,  $\rho$

**P2b** Rule  $\neg \vdash$ : If  $\lambda$ ,  $\rho \vdash \pi$ ,  $\phi$ , then  $\lambda$ ,  $\neg\phi$ ,  $\rho \vdash \pi$

**P3a** Rule  $\vdash \wedge$ : If  $\zeta \vdash \lambda$ ,  $\phi$ ,  $\rho$  and  $\zeta \vdash \lambda$ ,  $\psi$ ,  $\rho$ , then  $\zeta \vdash \lambda$ ,  $\phi \wedge \psi$ ,  $\rho$

**P3b** Rule  $\wedge \vdash$ : If  $\lambda$ ,  $\phi$ ,  $\psi$ ,  $\rho \vdash \pi$ , then  $\lambda$ ,  $\phi \wedge \psi$ ,  $\rho \vdash \pi$

**P4a** Rule  $\vdash \vee$ : If  $\zeta \vdash \lambda$ ,  $\phi$ ,  $\psi$ ,  $\rho$ , then  $\zeta \vdash \lambda$ ,  $\phi \vee \psi$ ,  $\rho$

**P4b** Rule  $\vee \vdash$ : If  $\lambda$ ,  $\phi$ ,  $\rho \vdash \pi$  and  $\lambda$ ,  $\psi$ ,  $\rho \vdash \pi$ , then  $\lambda$ ,  $\phi \vee \psi$ ,  $\rho \vdash \pi$

**P5a** Rule  $\vdash \rightarrow$ : If  $\zeta$ ,  $\phi \vdash \lambda$ ,  $\psi$ ,  $\rho$ , then  $\zeta \vdash \lambda$ ,  $\phi \rightarrow \psi$ ,  $\rho$

**P5b** Rule  $\rightarrow \vdash$ : If  $\lambda$ ,  $\psi$ ,  $\rho \vdash \pi$  and  $\lambda$ ,  $\rho \vdash \pi$ ,  $\phi$ , then  $\lambda$ ,  $\phi \rightarrow \psi$ ,  $\rho \vdash \pi$

**P6a** Rule  $\vdash \leftrightarrow$ : If  $\phi$ ,  $\zeta \vdash \lambda$ ,  $\psi$ ,  $\rho$  and  $\psi$ ,  $\zeta \vdash \lambda$ ,  $\phi$ ,  $\rho$ , then  $\zeta \vdash \lambda$ ,  $\phi \leftrightarrow \psi$ ,  $\rho$

**P6b** Rule  $\leftrightarrow \vdash$ : If  $\phi$ ,  $\psi$ ,  $\lambda$ ,  $\rho \vdash \pi$  and  $\lambda$ ,  $\rho \vdash \pi$ ,  $\phi$ ,  $\psi$ , then  $\lambda$ ,  $\phi \leftrightarrow \psi$ ,  $\rho \vdash \pi$

## Proofs

The basic idea in proving a sequent  $\pi \vdash \rho$  is to begin with instance(s) of Rule P1 and successively apply the remaining rules until you end up with the sequent you are hoping to prove.

For example, suppose you wanted to prove the sequent  $\neg(p \vee q) \vdash \neg p$ . One possible proof would proceed as follows.

1.  $p \vdash p, q$                       Rule 1
  2.  $p \vdash p \vee q$                     Rule P4a
  3.  $\vdash \neg p, p \vee q$                 Rule P2a
  4.  $\neg(p \vee q) \vdash \neg p$             Rule P2b
- QED.

However, a simpler idea (as it will involve much less search) is to begin with the sequent(s) to be proved and apply the rules above in the “backward” direction until you end up with the sequent you desire. In the example then, you would begin at Step 4 and apply each of the rules in the backward direction until you end up at Step 1 at which point you can conclude the original sequent is a theorem.

## Question Specification

In this assignment you are to emulate Hao Wang’s feats and implement a propositional theorem prover. You may use any programming language to complete this question. You must provide a script named `assn1q3` or a `Makefile` that, when the command `make` is executed, produces an executable file `assn1q3`.

### Input

The input will consist of a single sequent on the command line. Sequents will be written as: *[List of Formulae] seq [List of Formulae]* To construct formulae, atoms can be any string of characters (without space) and connectives as follows:

- $\neg$ : `neg`
- $\wedge$ : `and`
- $\vee$ : `or`
- $\rightarrow$ : `imp`
- $\leftrightarrow$ : `iff`

So, for example, the sequent  $p \rightarrow q, \neg r \rightarrow \neg q \vdash p \rightarrow r$  would be written as:

```
[p imp q, (neg r) imp (neg q)] seq [p imp r]
```

Your program should be called `assn1q3` and run as follows:

```
./assn1q3 'Sequent'
```

For example

```
./assn1q3 '[p imp q, (neg r) imp (neg q)] seq [p imp r]'
```

### Output

The first line of the output will be either `true` or `false` indicating whether or not the sequent on the command line holds. This output is worth 40% of the total mark for this question on given and hidden test data. The subsequent lines of output should produce a proof like the one in the *Proofs* section above.

## Marking for this Question

- Code: 40%
- Given test data: 20%
- Hidden test data: 20%
- Printing proofs: 20%

## References

- [1] Hao Wang, *Toward Mechanical Mathematics*, IBM Journal for Research and Development, volume 4, 1960. (Reprinted in: Hao Wang, "Logic, Computers, and Sets", Science Press, Peking, 1962. Hao Wang, "A Survey of Mathematical Logic", North Holland Publishing Company, 1964. Hao Wang, "Logic, Computers, and Sets", Chelsea Publishing Company, New York, 1970.)
- [2] Alfred North Whitehead and Bertrand Russell, *Principia Mathematica*, 2nd Edition, Cambridge University Press, Cambridge, England, 1927.

## A List of 10 Propositional Theorems

You may find it instructional to prove these by hand first.

- $\vdash \neg p \vee p$
- $\neg(p \vee q) \vdash \neg p$
- $p \vdash q \rightarrow p$
- $p \vdash p \vee q$
- $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$
- $p \leftrightarrow q \vdash \neg(p \leftrightarrow \neg q)$
- $p \leftrightarrow q \vdash (q \leftrightarrow r) \rightarrow (p \leftrightarrow r)$
- $\vdash (\neg p \wedge \neg q) \rightarrow (p \leftrightarrow q)$
- $p \leftrightarrow q \vdash (p \wedge q) \vee (\neg p \wedge \neg q)$
- $p \rightarrow q, \neg r \rightarrow \neg q \vdash p \rightarrow r$

### 4. [10 Marks] (Knowledge Representation and Reasoning)

Select a method for knowledge representation and reasoning that we have not covered in lectures and provide one example that explains:

- how the method represents knowledge;
- describes how inference works for reasoning with that knowledge representation.

In answering this question some sources you might consult include:

- Ronald Brachman and Hector Levesque, *Knowledge Representation and Reasoning*, Morgan Kaufmann, 2004.
- Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Third Edition, Prentice Hall, 2010.
- The *Principles of Knowledge Representation and Reasoning* conference series ([www.kr.org](http://www.kr.org)).
- The *Association for the Advancement of Artificial Intelligence* conference series ([www.ijcai.org](http://www.ijcai.org)).
- The *International Joint Conference on Artificial Intelligence* series ([www.aaai.org](http://www.aaai.org)).

## Assignment Submission

You will need to submit answers to Questions 1, 2, 4 and the report for Question 3 in a PDF file named `assn1.pdf` along with any source code files for Questions 3. Your report for Question 3 in `assn1.pdf` should describe the additional files you submit for this question and how they can be used to replicate/generate your results.

```
give cs4418 assn1 assn1.pdf assn1-q3-files
```

The deadline for this submission is 23:59:59am Sunday 13 October.

## Late Submissions

In case of late submissions, 10% will be deducted from the maximum mark for each day late.

No extensions will be given for any of the assignments (except in case of illness or misadventure). Read the course outline carefully for the rules regarding plagiarism.