COMP4418, 2021–Assignment 1

Due: 23:59:59pm Sunday 17 October (Week 5)
Late penalty: 10 marks per day.
Worth: 15%.

This assignment consi... The first two questions and the fourth question require written answers only. ...uires some programming and a written report.

[A] **[20 Marks]** (Lo...

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) $\forall x.P(x) \rightarrow Q(x), \forall x.Q(x) \rightarrow R(x), \neg R(a)[\models / \vdash]\neg P(a)$

[B] **[30 Marks]** (Logic Puzzle)

These problems are taken from Raymond M. Smullyan, *Logical Labyrinths*, A. K. Peters, 2009. The first two problems are trying to prepare you for the third which is more challenging.

> Edgar Abercrombie was an anthroplogist who was particularly interested in the logic and sociology of *lying* and *truth-telling*. One day he decided to visit a cluster of of islands where a lot of lying and truth-telling activity was going on! The first island of his visit was the Island of Knights and Knaves... in which all *knights* tell the truth and all *knaves* lie.

(i) Problem 12.1. On the first island he visited, all the inhabitants said the same thing: "All of us here are the same type."
What can be deduced about the inhabitants of that island?

(ii) Problem 12.3. On the next island, all the inhabitants said: "Some of us are knights and some are knaves."
What is the composition of the island?

(iii) Problem 12.14. On the enxt island visited by Abercrombie, he met six natives, named Arthur, Bernard, Charles, David, Edward, and Frank, who made the following statements:
Arthur: Everyone here smokes cigarettes.
Bernard: Everyone here smokes cigars.
Charles: Everyone here smokes either cigarettes or cigars or both.
David: Arthur and Bernard are not both knaves.
Edward: If Charles is a knight, so is David.
Frank: If David is a knight, so is Charles.

Is it possible to determine of any one of these that he is a knight, and if so, which one or ones?

For each problem:

(i) Represent the facts in the paragraph in first-order logic.

(ii) Using your formalisation in part (i), is it possible to answer the question? Show semantically how you determined your answer.

(iii) If your answer to part (ii) was 'no', indicate what further sentences you would need to add to your formal[illegible]ld conclude the answer to the question (i.e., whether C is a knight or [illegible]

(iv) Using all th[illegible]dded, determine the answer to the question.

[C] [**30 Marks**] (Au[illegible]ving)

In 1958 the logi[illegible]mented one of the first automated theorem provers. He succeeded in wri[illegible] capable of automatically proving a majority of theorems from the first five[illegible]d 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 π and ρ are strings (i.e., lists) of formulae, can be read as "the formulae in the string ρ *follow* from the formulae in the string π" (or, equivalently, "the formulae in string π *prove* the formulae in string ρ").

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 ϕ 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, $\ldots$) is an *atomic* formula (and thus a formula).
- If ϕ, ψ are formulae, then $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$ are formulae.

Sequent

If π and ρ are strings of formulae (possibly empty strings) and ϕ is a formula, then π, ϕ, ρ 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 λ, ζ 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 λ and ζ 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 $\neg \vdash$: [illegible] $\neg\phi,\ \rho \vdash \pi$

P3a Rule $\vdash \wedge$: [illegible] $\lambda,\ \psi,\ \rho$, then $\zeta \vdash \lambda,\ \phi \wedge \psi,\ \rho$

P3b Rule $\wedge \vdash$: [illegible] $\lambda,\ \phi \wedge \psi,\ \rho \vdash \pi$

P4a Rule $\vdash \vee$: [illegible] $\zeta \vdash \lambda,\ \phi \vee \psi,\ \rho$

P4b Rule $\vee \vdash$: [illegible] $,\ \rho \vdash \pi$, then $\lambda,\ \phi \vee \psi,\ \rho \vdash \pi$

P5a Rule $\vdash \rightarrow$: [illegible] $\zeta \vdash \lambda,\ \phi \rightarrow \psi,\ \rho$

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 $p \to q,\ \neg r \to \neg q \vdash p \to r$ 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 [illegible]er `true` or `false` indicating whether or not the sequent on the command li[illegible] is worth 40% of the total mark for this question on given and hidden test [illegible] 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", Sciene 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.

(i) $\vdash \neg p \vee p$

(ii) $\neg(p \vee q) \vdash \neg p$

(iii) $p \vdash q \to p$

(iv) $p \vdash p \vee q$

(v) $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$

(vi) $p \leftrightarrow q \vdash \neg(p \leftrightarrow \neg q)$

(vii) $p \leftrightarrow q \vdash (q \leftrightarrow r) \to (p \leftrightarrow r)$

(viii) $\vdash (\neg p \wedge \neg q) \to (p \leftrightarrow q)$

(ix) $p \leftrightarrow q \vdash (p \wedge q) \vee (\neg p \wedge \neg q)$

(x) $p \to q,\ \neg r \to \neg q \vdash p \to r$

[D] [**20 Marks**] (Knowledge Representation and Reasoning)

Download the article: Ernest Davis and Gary Marcus, *Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence*, Communications of the ACM, **58**(9):92–103, September 2015.

You can access it from: `https://doi.org/10.1145/2701413`.

You can also access it via the UNSW Library searching on the title “Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence.” If you have problems accessing the article, please post on the COMP4418 Forum so that we can help you locate and download this article.

In 1-page at most you should:

- Provide a brief summary (at most half a page) of the article and summarise the main points made in this paper;
- Provide comments on:
  - 1 point made in the paper with which you agree and explain why?
  - 1 point [illegible] h which you consider questionable and explain why?

Assignment S[illegible]

You will need to subm[illegible] ons 1, 2, 4 and the report for Question 3 in a PDF file named `assn1.pdf` alo[illegible] ode files for Questions 3. Your report for Question 3 in `assn1.pdf` should des[illegible] es you submit for this question and how they can used to replicate/generate your results.

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

The deadline for this submission is 23:59:59am Sunday 17 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.