COMP4418, 2018 – Assignment 2

Due: 14:59:59pm 01 October (Week 10)
Late penalty: 10 marks per day

Worth: 15 %

This assignment consists of seven questions.

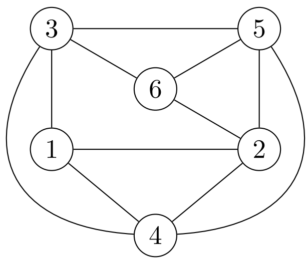
1. **[15 Marks]** (Answer Set Programming)
A *clique* of a graph is a set of vertices C that are pairwise adjacent. That is, for a graph with vertex set V and set of undirected edges $E \subseteq \{\{x, y\} \mid x, y \in V\}$, a clique is a set $C \subseteq V$ such that for all $x \in C$ and $y \in C$, if $x \neq y$, then $\{x, y\} \in E$. A k-*clique* is a clique C of size k, that is, $|C| = k$.

   (a) Write an ASP program that decides the k-Clique problem:

   Input: a graph with vertices V and edges $E \subseteq V \times V$, and a natural number $k \geq 0$.
   Problem: decide if there is a k-clique.

   Assume the input parameters V, E, k are encoded in ASP in the form of a unary predicate `v`, a binary predicate `e`, and a constant symbol `k`, respectively. Use a unary predicate `c` to represent the clique C. Your program should have no more than two rules.

   (b) Use an ASP solver[1] to determine how many k-cliques the following graph has for every $k \in \{3, 4, 5, 6\}$.
   Hint: the number of 2-cliques is 11.



2. **[15 Marks]** (Answer Set Programming)
Consider the following logic program P.

$$
\begin{aligned}
a &\leftarrow \text{not}\, b, \text{not}\, c. \\
b &\leftarrow \text{not}\, a, \text{not}\, c. \\
c &\leftarrow \text{not}\, a, \text{not}\, b. \\
d &\leftarrow a. \\
d &\leftarrow b. \\
d &\leftarrow c.
\end{aligned}
$$

Determine stable models of this program. For every candidate interpretation S, specify the reduct P^S. Give your solution in a table of exactly (!) the following form and order:

[1] For instance, you can download the Clingo ASP solver from `https://potassco.org/` and run your program with `clingo --const k=`k `-n 0 clique.lp` or `./clingo --const k=`k `-n 0 clique.lp` where $k \in \mathbb{N}$ is the size of the clique.

S	Reduct P^S	Stable model?
$\{a, b, c, d\}$	$d \leftarrow a.\quad d \leftarrow b.\quad d \leftarrow c.$	✗
$\{a, b, c\}$		
$\{a, b, d\}$		
$\{a, c, d\}$		
$\{b, c, d\}$		
$\{a, b\}$		
$\{a, c\}$		
$\{a, d\}$		
$\{b, c\}$		
$\{b, d\}$		
$\{c, d\}$		
$\{a\}$		
$\{b\}$		
$\{c\}$		
$\{d\}$		
$\{\}$		

3. **[5 Marks]** (Satisfiability)
Download the following material from the course website:

- Lectures → Week 6 & 7 → `sat-naive.cc` (implements Algorithm 1)
- Lectures → Week 6 & 7 → `sat-up.cc` (implements Algorithm 2)
- Lectures → Week 6 & 7 → `sat-cdcl.cc` (implements Algorithm 3)
- Assignments → `sudoku.cnf` (CNF encoding of a Sudoku game)

On the CSE machines, you can compile the files by running the following commands:

```
$ c++ -std=c++11 -O3 -DNDEBUG -o sat-naive sat-naive.cc
$ c++ -std=c++11 -O3 -DNDEBUG -o sat-up sat-up.cc
$ c++ -std=c++11 -O3 -DNDEBUG -o sat-cdcl sat-cdcl.cc
```

This will create three executable files: `sat-naive`, `sat-up`, `sat-cdcl`. Use them to solve the Sudoku instance `sudoku.cnf`:

```
$ ./sat-naive sudoku.cnf
$ ./sat-up sudoku.cnf
$ ./sat-cdcl sudoku.cnf
```

(If the solver does not report a result after 5 minutes, you can terminate the process.)

(a) What are the run times reported by the individual solvers?
(b) Briefly explain why the run times differ (two or three sentences).

4. **[18 Marks]** (Satisfiability)
State for each of the following statements whether it is true. Briefly explain your answer in one or two sentences.

(a) Converting a propositional formula into an equisatisfiable CNF formula in the worst case requires exponential time (under the assumption P $\neq$ NP).
(b) There are decision problems that cannot be reduced to SAT (if so, name a concrete problem).
(c) If I is closed under unit propagation relative to ϕ, then in order to close $I \cup \{x\}$ under unit propagation relative to ϕ it suffices to inspect the clauses $c \in \phi$ that watch $\bar{x}$.

5. **[15 Marks]** (Satisfiability)
Consider the CNF formula ϕ

$$(p \vee q \vee r \vee s) \wedge (p \vee \bar{q} \vee \bar{t}) \wedge (p \vee t)$$

and the partial interpretation $I = \{\bar{p}\}$. Close I under unit propagation relative to ϕ using the watched-literal scheme. Initially, the clauses shall be watching the following literals:

- $p \vee q \vee r \vee s$ watches p and q,
- $p \vee \bar{q} \vee \bar{t}$ watches p and $\bar{q}$,
- $p \vee t$ watches p and t.

Use a table of the following form to show how I and the watched literals evolve. In each row, use the cell below a clause to list the literals currently watched by the clause, and when a new literal is derived by unit propagation, add it to the next free cell in the first column.

I	Clauses and Watched Literals		
	$p \vee q \vee r \vee s$	$p \vee \bar{q} \vee \bar{t}$	$p \vee t$
	p, q	$p, \bar{q}$	p, t
$\bar{p}$			

Hint: the closure of $\{\bar{p}\}$ under unit propagation relative to ϕ is $\{\bar{p}, \bar{q}, t\}$, so a table with three rows like the one below should suffice.

6. **[12 Marks]** (Reasoning about Knowledge)
For each of the the following statements, prove whether it is valid, or not valid but satisfiable, or unsatisfiable. That is, to prove that ϕ is valid, you need to show that all interpretations e, w satisfy ϕ; to prove that ϕ is satisfiable, you need to show that some interpretation e, w satisfies ϕ; to prove that ϕ is unsatisfiable, you need to prove that no interpretation e, w satisfies ϕ.

   For instance, one way to prove validity of $(\mathbf{K}\,\phi \wedge \mathbf{K}\,(\phi \rightarrow \psi)) \rightarrow \mathbf{K}\,\psi$ is as follows: Let e, w be an arbitrary interpretation. Suppose $e, w \models \mathbf{K}\,\phi \wedge \mathbf{K}\,(\phi \rightarrow \psi)$; we need to show $e, w \models \mathbf{K}\,\psi$. Then for all $e, w \models \mathbf{K}\,\phi$ and $e, w \models \mathbf{K}\,(\phi \rightarrow \psi)$. The former means that for all $w' \in e$, $e, w' \models \phi$; the latter means that for all $w' \in e$, $e, w' \models (\phi \rightarrow \psi)$. Therefore for all $w' \in e$, $e, w' \models \phi$ and $e, w' \models (\phi \rightarrow \psi)$, and therefore $e, w' \models \psi$. Thus for all $w' \in e$, $e, w' \models \psi$. Hence $e, w \models \mathbf{K}\,\psi$.

   (a) $\mathbf{K}\,\text{Happy} \wedge \mathbf{K}\,\neg\text{Happy}$

   (b) $\mathbf{K}\,(\text{Happy} \vee \text{Sad}) \rightarrow \neg\mathbf{K}\,\text{Happy}$

   You may abbreviate the propositions by their first letters.

7. **[20 Marks]** (Reasoning about Action & Change)
In the *blocks world* a robot is stacking blocks on a table. We formalised this scenario using STRIPS in the lecture. Here, your task is to represent the actions and their effects from the *blocks world* using the Logic of Actions. For this purpose you should familiarise yourself with the actions and propositions in the STRIPS model first.

   The limitations of STRIPS required us to introduce some redundancies in the STRIPS model. When translating the STRIPS model to the Logic of Actions, some of these redundancies can be avoided. For example, we can simplify the actions in the following way.

   - The action $\text{pickUp}(x)$ replaces the STRIPS action $\text{pickUp}(x, y)$ because we can figure out in the successor-state axiom on which table or block y the block x is.
   - The action $\text{putOn}(y)$ replaces the STRIPS action $\text{putOn}(x, y)$ because we can figure out in the successor-state axiom which block x the robot is holding.
   - The action putOnTable replaces the STRIPS action $\text{putOnTable}(x)$ because we can figure out in the successor-state axiom which x the robot is holding.

With these simplified actions, we can capture the behaviour of the predicate $\mathrm{On}(x, y)$ with the following successor-state axiom:

$$\Box\forall a\forall x\forall y\Big([a]\mathrm{On}(x,y) \leftrightarrow \big(\mathrm{Holding}(x) \wedge (a = \mathrm{putOn}(y) \vee (y = \mathrm{T} \wedge a = \mathrm{putOnTable}))\big) \vee$$
$$\big(\mathrm{On}(x,y) \wedge a \neq \mathrm{pickUp}(x)\big)\Big).$$

(a) Similarly to the successor-state axiom given above, provide a successor-state axiom for the predicate $\mathrm{Holding}(x)$ that captures the meaning of $\mathrm{Holding}(x)$ from the STRIPS model. That is, your successor-state axiom should capture the effects of actions like $\mathrm{pickUp}(y)$ and $\mathrm{putOn}(y)$ on which x the robot is holding.

(b) Show that after picking up B and putting it on C, B is on C, using regression. That is, regress the formula

$$[\mathrm{pickUp(B)}][\mathrm{putOn(C)}]\mathrm{On(B,C)}$$

and prove that the resulting formual is valid.

Please make sure that your solution is readable. Show intermediate steps. To keep the presentation concise, you can abbreviate $\mathrm{Holding}(x)$ by $\mathrm{H}(x)$, $\mathrm{pickUp}(x)$ by $\mathrm{pu}(x)$, $\mathrm{putOn}(x)$ by $\mathrm{po}(x)$, and $\mathrm{putOnTable}(x)$ by $\mathrm{pot}(x)$. Moreover, you may make simplifying steps when subformulas are obviously false/true; for instance, for the formula

$$\mathrm{H}(C) \wedge (\mathrm{po}(C) = \mathrm{po}(C) \vee (\mathrm{C} = \mathrm{T} \wedge \mathrm{po}(C) = \mathrm{pot}))$$

it is immediate that $\mathrm{C} = \mathrm{T} \wedge \mathrm{po}(C) = \mathrm{pot}$ is trivially false, whereas $\mathrm{po}(C) = \mathrm{po}(C)$ is trivially true, so the whole formula can be simplified to $\mathrm{H}(C)$.

(c) The STRIPS formalisation mentions the proposition schema $\mathrm{Clear}(x)$. Is such a predicate also needed when using the Logic of Action to model the *blocks world*, or is it redundant? Explain your answer. If $\mathrm{Clear}(x)$ is redundant, provide a formula that expresses its meaning using the predicates $\mathrm{On}(x, y)$ and/or $\mathrm{Holding}(x)$.

Submission

You will need to answer the questions in a file named `assn2.pdf` Submit using the command:

```
give cs4418 assn2 assn2.pdf
```

The deadline for this submission is 14:59:59pm 01 October.

Late Submissions

In case of late submissions, the maximum available mark is reduced by 10 points for each day late.

No extensions will be given for the assignment (except in case of illness or misadventure).

Academic Honesty and Plagiarism

All work submitted for assessment **must be your own work**. Assignments **must be completed individually**. We regard copying of assignments, in whole or part, as a very serious offence. Be warned that:

- the submission of work derived from another person, or jointly written with someone else will, at the very least, result in automatic failure for COMP4418 with a mark of zero;
- allowing another student to copy from you will, at the very least, result in a mark of zero for your own assignment; and
- severe or second offences will result in automatic failure, exclusion from the University, and possibly other academic discipline.

Collaborative work in the form of "think tanking" is encouraged, but students are not allowed to derive solutions together as a group during such discussions. Students are also warned not to send solution fragments of the assignments to each other in any form (e.g. as email or listings). In addition, copying/purchasing of solutions that is available on the web is also not permitted. Students are strongly advised to protect their work. Do not leave your terminal/computer unattended, or leave printouts at the printer for others to take. Read the study guide carefully for the rules regarding plagiarism.