Due: 1st October 2017 at 11:59pm

COMP 9020 – Assignment 2

1. Eight houses are lined up on a street, with four on each side of the road as shown:

Each house wants to set up its own wi-fi network, but the wireless networks of neighbouring houses – that is, houses that are either next to each other (ignoring trees) or over the road from one another (directly opposite) – can interfere, and must therefore be on different channels. Houses that are sufficiently far away may use the same wi-fi channel. Your goal is to find the minimum number of different channels the neighbourhood requires.

   (a) Explain how this can be formulated as a graph-based problem.

   (b) What is the minimum number of wi-fi channels required for the neighbourhood?

   (c) How does your answer to (b) change if a house's wireless network can also interfere with those of the houses to the left and right of the house over the road?

2. In the lectures it was mentioned that determining if a graph can be three-coloured is a "difficult problem". It was also mentioned that determining if a formula in CNF was satisfiable is a "difficult problem". We will now show the two problems are related, relatively simply, by *reducing* the three-colourability problem to the satisfiability problem. That is, given a graph, G, we will define a formula (in CNF) φ_G such that G is three-colourable if, and only if φ_G is satisfiable.

Let $G = (V, E)$ be a graph, and $C = \{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$ be a set of three colours. Define $\textsc{Prop} := \{P_{v,c} \ : \ v \in V \text{ and } c \in C\}$. Intuitively, $P_{v,c}$ represents the proposition "vertex v has colour c". Recall that a *literal* is either p or $\neg p$ where $p \in \textsc{Prop}$.

   (a) Define the proposition A_v: "vertex v has at least one colour" using a disjunction of literals.

   (b) Define the proposition B_v: "vertex v has at most one colour" in CNF.

(c) Define the proposition $C_{u,v}$: "vertex u and vertex v have different colours" in CNF.

(d) Define φ_G. You may wish to use the notation $\bigwedge_{i=0}^{k} \psi_i$, which is shorthand for $\psi_0 \wedge \psi_1 \wedge \ldots \wedge \psi_k$, though the variations $\bigwedge_{v \in V}$ and $\bigwedge_{\{u,v\} \in E}$ will likely be more useful.

3. Let $(A, \vee, \wedge, ', 0, 1)$ be a Boolean algebra. Define a relation $\sqsubseteq$ on A as follows:

$$x \sqsubseteq y \quad \text{if} \quad x \vee y = y.$$

(a) Show that $\sqsubseteq$ is a partial order (using the laws of Boolean algebra).

(b) In the Boolean algebra defined by the subsets of a set X, what partial order does $\sqsubseteq$ correspond to?

(c) While $\sqsubseteq$ is not very interesting in the two-valued logic associated with Propositional Logic, the "propositional analogue" would be the Boolean function:

$$(x \vee y) \leftrightarrow y.$$

What Boolean connective is $(x \vee y) \leftrightarrow y$ logically equivalent to?

4. We can define addition[1] over natural numbers inductively as follows:

- $\mathsf{add}(m, 0) = m$, and
- $\mathsf{add}(m, n+1) = \mathsf{add}(m, n) + 1$.

We will now show that add is commutative!

(a) Let $P(n)$ be the proposition

$$P(n): \ \mathsf{add}(n, 0) = \mathsf{add}(0, n).$$

Prove that $P(n)$ holds for all $n \in \mathbb{N}$.

(b) Let $Q(n)$ be the proposition

$$Q(n): \text{ If } a + b = n \text{ and } a, b > 0 \text{ then } \mathsf{add}(a, b) = \mathsf{add}(b, a).$$

Prove that $Q(n)$ holds for all $n \in \mathbb{N}$. (*Hint*: It may be easier to show that $P(n) \wedge Q(n)$ holds for all n).

5. Define the sequence, a_n, recursively as:

$$a_0 = 0 \qquad a_1 = 1 \qquad a_n = 5a_{n-1} - 6a_{n-2}.$$

[1] Note: the "+1" in "n+1" should be considered a structural concept, and not (immediately) related to the function we are defining

Consider the following algorithms for computing a_n:

```
rec_a(n) :
  if n < 2 :  return n
  else :
    x := rec_a(n − 1)
    y := rec_a(n − 2)
    return 5x − 6y
```

```
iter_a(n) :
  if n < 2 :  return n
  else :
    x := 1
    y := 0
    i := 1
    while i < n :
      t := x
      x := 5x − 6y
      y := t
      i := i + 1
    return x
```

(a) Give asymptotic upper bounds for the running times for rec_a and iter_a to compute a_n.

(b) By considering $a_n + 2^n$, guess an expression for a_n. Prove your guess is correct for all $n \in \mathbb{N}$.

(c) Can you find a more efficient (i.e. asymptotically faster) method for computing a_n?

Advice on how to do the assignment

All submitted work must be done individually without consulting someone else's solutions in accordance with the University's "Academic Dishonesty and Plagiarism" policies.

- Assignments are to be submitted via WebCMS (or give) as a pdf
- Be careful with giving multiple or alternative answers. If you give multiple answers, then we will give you marks only for "your worst answer", as this indicates how well you understood the question.
- Some of the questions are very easy (with the help of the lecture notes or book). You can use the material presented in the lecture or book (without proving it). You do not need to write more than necessary (see comment above).
- When giving answers to questions, we always would like you to prove/explain/motivate your answers.
- If you use further resources (books, scientific papers, the internet,...) to formulate your answers, then add references to your sources.