Course MA2C01: Michaelmas Term 2010.

Assignment II.

Worked solutions.

1. (a) *Let $*$ denote the binary operation on the set $\mathbb{R}^3$ of ordered triples of real numbers defined such that*

$$\begin{aligned}&(a_1, a_2, a_3) * (b_1, b_2, b_3)\\ &\quad = \Bigl(a_1b_3 + a_2b_2 + a_3b_1,\;\; a_1b_1 + a_2b_3 + a_3b_2,\;\; a_1b_2 + a_2b_1 + a_3b_3\Bigr).\end{aligned}$$

Prove that $\mathbb{R}^3, *)$ is a monoid. Is this monoid a group? [Justify your answers.]

Let (a_1, a_2, a_3) (b_1, b_2, b_3) and (c_1, c_2, c_3) be elements of $\mathbb{R}^3$. Then

$$\begin{aligned}
&((a_1, a_2, a_3) * (b_1, b_2, b_3)) * (c_1, c_2, c_3)\\
&\quad = \Bigl(a_1b_3 + a_2b_2 + a_3b_1,\;\; a_1b_1 + a_2b_3 + a_3b_2,\;\; a_1b_2 + a_2b_1 + a_3b_3\Bigr)\\
&\qquad\quad * (c_1, c_2, c_3)\\
&\quad = \Bigl((a_1b_3 + a_2b_2 + a_3b_1)c_3 + (a_1b_1 + a_2b_3 + a_3b_2)c_2\\
&\qquad\qquad + (a_1b_2 + a_2b_1 + a_3b_3)c_1,\\
&\qquad\quad (a_1b_3 + a_2b_2 + a_3b_1)c_1 + (a_1b_1 + a_2b_3 + a_3b_2)c_3\\
&\qquad\qquad + (a_1b_2 + a_2b_1 + a_3b_3)c_2,\\
&\qquad\quad (a_1b_3 + a_2b_2 + a_3b_1)c_2 + (a_1b_1 + a_2b_3 + a_3b_2)c_1\\
&\qquad\qquad + (a_1b_2 + a_2b_1 + a_3b_3)c_3\Bigr),\\
&\quad = \Bigl(a_1b_3c_3 + a_2b_2c_3 + a_3b_1c_3 + a_1b_1c_2 + a_2b_3c_2 + a_3b_2c_2\\
&\qquad\qquad + a_1b_2c_1 + a_2b_1c_1 + a_3b_3c_1,\\
&\qquad\quad a_1b_3c_1 + a_2b_2c_1 + a_3b_1c_1 + a_1b_1c_3 + a_2b_3c_3 + a_3b_2c_3\\
&\qquad\qquad + a_1b_2c_2 + a_2b_1c_2 + a_3b_3c_2,\\
&\qquad\quad a_1b_3c_2 + a_2b_2c_2 + a_3b_1c_2 + a_1b_1c_1 + a_2b_3c_1 + a_3b_2c_1\\
&\qquad\qquad + a_1b_2c_3 + a_2b_1c_3 + a_3b_3c_3\Bigr),
\end{aligned}$$

and

$$\begin{aligned}
&(a_1, a_2, a_3) * ((b_1, b_2, b_3) * (c_1, c_2, c_3))\\
&\quad = (a_1, a_2, a_3) *
\end{aligned}$$

$$
\begin{aligned}
&= \Big(b_1c_3 + b_2c_2 + b_3c_1,\;\; b_1c_1 + b_2c_3 + b_3c_2,\;\; b_1c_2 + b_2c_1 + b_3c_3\Big)\\
&= \Big(a_1(b_1c_2 + b_2c_1 + b_3c_3) + a_2(b_1c_1 + b_2c_3 + b_3c_2)\\
&\qquad + a_3(b_1c_3 + b_2c_2 + b_3c_1),\\
&\quad a_1(b_1c_3 + b_2c_2 + b_3c_1) + a_2(b_1c_2 + b_2c_1 + b_3c_3)\\
&\qquad + a_3(b_1c_1 + b_2c_3 + b_3c_2),\\
&\quad a_1(b_1c_1 + b_2c_3 + b_3c_2) + a_2(b_1c_3 + b_2c_2 + b_3c_1)\\
&\qquad + a_3(b_1c_2 + b_2c_1 + b_3c_3)\Big)\\
&= \Big(a_1b_1c_2 + a_1b_2c_1 + a_1b_3c_3 + a_2b_1c_1 + a_2b_2c_3 + a_2b_3c_2\\
&\qquad + a_3b_1c_3 + a_3b_2c_2 + a_3b_3c_1,\\
&\quad a_1b_1c_3 + a_1b_2c_2 + a_1b_3c_1 + a_2b_1c_2 + a_2b_2c_1 + a_2b_3c_3\\
&\qquad + a_3b_1c_1 + a_3b_2c_3 + a_3b_3c_2,\\
&\quad a_1b_1c_1 + a_1b_2c_3 + a_1b_3c_2 + a_2b_1c_3 + a_2b_2c_2 + a_2b_3c_1\\
&\qquad + a_3b_1c_2 + a_3b_2c_1 + a_3b_3c_3\Big)\\
&= ((a_1, a_2, a_3) * (b_1, b_2, b_3)) * (c_1, c_2, c_3)
\end{aligned}
$$

It follows that the binary operation $*$ on $\mathbb{R}^3$ is associative. Now

$$(a_1, a_2, a_3) * (0, 0, 1) = (a_1, a_2, a_3)$$

and

$$(0, 0, 1) * (a_1, a_2, a_3) = (a_1, a_2, a_3)$$

for all $(a_1, a_2, a_3) \in \mathbb{R}^3$. Therefore $(0, 0, 1)$ is an identity element for the binary operation $*$ on $\mathbb{R}^3$. We have thus shown that $(\mathbb{R}^3, *)$ is a monoid.

Note that

$$(1, 1, 1) * (b_1, b_2, b_3) = (c, c, c)$$

for all $(b_1, b_2, b_3) \in \mathbb{R}^3$, where $c = b_1 + b_2 + b_3$. It follows that there cannot exist any element (b_1, b_2, b_3) of $\mathbb{R}^3$ for which $(1, 1, 1)*(b_1, b_2, b_3) = (0, 0, 1)$. It follows that the element $(1, 1, 1)$ of $\mathbb{R}^3$ is not an invertible element of this monoid. Therefore the monoid is not a group.

(b) *Let $f\colon \mathbb{R}^3 \to \mathbb{C}$ be the function defined such that*

$$f(a_1, a_2, a_3) = a_3 - \frac{1}{2}(a_1 + a_2) + \frac{\sqrt{3}}{2}(a_1 - a_2)i$$

for all $a_1, a_2, a_3 \in \mathbb{R}$, where $i^2 = -1$. Prove that f is a homomorphism between the monoids $(\mathbb{R}^3, *)$ and $(\mathbb{C}, \times)$, where $\times$ denotes the standard multiplication operation on the set $\mathbb{C}$ of complex numbers.

$$
\begin{aligned}
&f((a_1, a_2, a_3) * (b_1, b_2, b_3)) \\
&\quad = f\Big(a_1b_3 + a_2b_2 + a_3b_1,\ \ a_1b_1 + a_2b_3 + a_3b_2,\ \ a_1b_2 + a_2b_1 + a_3b_3\Big) \\
&\quad = a_1b_2 + a_2b_1 + a_3b_3 \\
&\qquad\quad - \frac{1}{2}(a_1b_3 + a_2b_2 + a_3b_1 + a_1b_1 + a_2b_3 + a_3b_2) \\
&\qquad\quad + \frac{\sqrt{3}}{2}(a_1b_3 + a_2b_2 + a_3b_1 - a_1b_1 - a_2b_3 - a_3b_2)i
\end{aligned}
$$

and

$$
\begin{aligned}
&f(a_1, a_2, a_3) * f(b_1, b_2, b_3) \\
&\quad = \left(a_3 - \frac{1}{2}(a_1 + a_2) + \frac{\sqrt{3}}{2}(a_1 - a_2)i\right) \\
&\qquad\quad \times \left(b_3 - \frac{1}{2}(b_1 + b_2) + \frac{\sqrt{3}}{2}(b_1 - b_2)i\right) \\
&\quad = a_3b_3 - \frac{1}{2}(a_3b_1 + a_3b_2 + a_1b_3 + a_2b_3) \\
&\qquad\quad + \frac{1}{4}(a_1 + a_2)(b_1 + b_2) - \frac{3}{4}(a_1 - a_2)(b_1 - b_2) \\
&\qquad\quad + \frac{\sqrt{3}}{2}\Big(a_3(b_1 - b_2) + (a_1 - a_2)b_3\Big)i \\
&\qquad\quad - \frac{\sqrt{3}}{4}\Big((a_1 + a_2)(b_1 - b_2) + (a_1 - a_2)(b_1 + b_2)\Big)i \\
&\quad = a_3b_3 - \frac{1}{2}(a_3b_1 + a_3b_2 + a_1b_3 + a_2b_3) \\
&\qquad\quad + \frac{1}{4}(a_1b_1 + a_1b_2 + a_2b_1 + a_2b_2) \\
&\qquad\quad - \frac{3}{4}(a_1b_1 - a_1b_2 - a_2b_1 + a_2b_2) \\
&\qquad\quad + \frac{\sqrt{3}}{2}(a_3b_1 - a_3b_2 + a_1b_3 - a_2b_3)i \\
&\qquad\quad - \frac{\sqrt{3}}{2}(a_1b_1 - a_2b_2)i
\end{aligned}
$$

$$\begin{aligned} &= a_3b_3 + a_1b_2 + a_2b_1 - \frac{1}{2}(a_3b_1 + a_3b_2 + a_1b_3 + a_2b_3 + a_1b_1 + a_2b_2) \\ &\quad + \frac{\sqrt{3}}{2}(a_3b_1 + a_1b_3 + a_2b_2 - a_3b_2 - a_2b_3 - a_1b_1)i \\ &= f((a_1, a_2, a_3) * (b_1, b_2, b_3)) \end{aligned}$$

Thus $f\colon \mathbb{R}^3 \to \mathbb{C}$ is a homomorphism.

2. (a) *Describe the language over the alphabet* $\{0, 1\}$ *generated by the context-free grammar whose non-terminals are* $\langle S\rangle$ *and* $\langle A\rangle$*, whose start symbol is* $\langle S\rangle$ *and whose productions are*

$$\langle S\rangle \to \langle S\rangle\langle A\rangle, \quad \langle S\rangle \to 1, \quad \langle A\rangle \to 01.$$

Is the context-free grammar a regular grammar?

The language generated by this grammar consists of those strings

$$1, \; 101, \; 10101, \; 1010101, \ldots$$

of binary digits in which the digits 0 and 1 alternate, and which start and end with the digit 1.

A typical derivation of one of these strings from the grammar is the following:

$$\begin{aligned} \langle S\rangle &\Rightarrow \langle S\rangle\langle A\rangle \Rightarrow \langle S\rangle\langle A\rangle\langle A\rangle \Rightarrow \langle S\rangle\langle A\rangle\langle A\rangle\langle A\rangle \\ &\Rightarrow \langle S\rangle\langle A\rangle\langle A\rangle 01 \Rightarrow \langle S\rangle\langle A\rangle 0101 \Rightarrow \langle S\rangle 010101 \Rightarrow 1010101 \end{aligned}$$

This grammar is not a regular grammar. The production $\langle S\rangle \to \langle S\rangle\langle A\rangle$ does not match any of the forms allowed for a production in a regular grammar.

(b) *Let* L *be the language over the alphabet* $\{0, 1\}$ *consisting of those finite strings of binary digits in which neither* 010 *nor* 101 *occurs as a substring. Give the description of a finite state acceptor for the language* L*, specifying the starting state, the finishing state or states, and the transition table for this finite state acceptor.*

The finite state acceptor specification is as follows:

- internal states: S, A, B, C, D, E
- start state: S
- finishing states: S, A, B, C, D
- transition table:

	0	1
S	A	B
A	A	C
B	D	B
C	E	B
D	A	E
E	E	E

(The empty string is considered as one of the finite strings in the language L. If only non-empty strings are allowed, omit the state S from the list of finishing states.)

The machine is placed in one of the states A and D at any stage where the last digit input is 0 and no error has occurred. The machine is placed in one of the states B and C at any stage where the last digit input is 1 and no error has occurred. The machine is placed in state C at any stage where the last two digits input are 01 and no error has occurred. If the next input digit is 0 then the machine is placed in the error state E, and cannot subsequently leave this error state to arrive at any finishing state. The machine is placed in state D at any stage where the last two digits input are 10 and no error has occurred. If the next input digit is 1 then the machine is placed in the error state E, and cannot subsequently leave this error state to arrive at any finishing state.

(c) *Construct a regular context-free grammar that generates the langle L described in (b).*

Specification of regular context-free grammar is as follows:

- terminals: 0, 1
- nonterminals: $\langle S \rangle$, $\langle A \rangle$, $\langle B \rangle$, $\langle C \rangle$, $\langle D \rangle$,
- start symbol: $\langle S \rangle$,
- productions:

$$\begin{aligned}
\langle S \rangle &\rightarrow 0\langle A \rangle \\
\langle S \rangle &\rightarrow 1\langle B \rangle \\
\langle A \rangle &\rightarrow 0\langle A \rangle \\
\langle A \rangle &\rightarrow 1\langle C \rangle \\
\langle B \rangle &\rightarrow 0\langle D \rangle \\
\langle B \rangle &\rightarrow 1\langle B \rangle \\
\langle C \rangle &\rightarrow 1\langle B \rangle
\end{aligned}$$

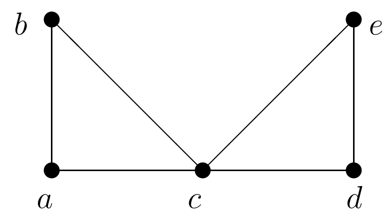
$$\begin{aligned}
\langle D\rangle &\rightarrow 0\langle A\rangle \\
\langle S\rangle &\rightarrow \varepsilon \\
\langle A\rangle &\rightarrow \varepsilon \\
\langle B\rangle &\rightarrow \varepsilon \\
\langle C\rangle &\rightarrow \varepsilon \\
\langle D\rangle &\rightarrow \varepsilon
\end{aligned}$$

(where ε denotes the empty string).

3. *Answer the following questions concerning the graph G with vertices V and edges E, where*

$$\begin{aligned}
V &= \{a,\, b,\, c,\, d\;\, e\} \\
E &= \{a\,b,\;\; a\,c,\;\; b\,c,\;\; c\,d,\;\; c\,e,\;\; d\,e\}.
\end{aligned}$$

[Briefly justify all your answers.]



(a) *Is the graph complete?*

No. There is no edge be, for example. It is not true that each pair of distinct vertices are adjacent (i.e., joined by a single edge).

(b) *Is the graph regular?*

No. The vertices do not all have the same degree. Vertices a, b, d and e have degree 2, whereas vertex c has degree 4.

(c) *Is the graph connected?*

Yes. All vertices are adjacent to the vertex c, and therefore any two vertices may be joined by a path of length at most two.

(d) *Does the graph have an Eulerian circuit?*

The graph is non-trivial and connected, and all the vertices are of even degree. Therefore an Eulerian circuit should exist. One such is $a\,b\,c\,d\,e\,c\,a$. (This circuit traverses every edge exactly once, and is thus an Eulerian circuit.)

(e) *Does the graph have a Hamiltonian circuit?*

No. Any circuit starting and ending at the vertex a (say) and passing through vertices b, d (and e) must pass through the vertex c at least twice, since all walks from a or b to d or e pass through the vertex c. Indeed if a circuit starts at the vertex a, it must pass through c to get to vertices d and e, and it must then return again through c in order to get back to a. Therefore there is no circuit that passes through all vertices of the graph and in addition passes through the vertex c exactly once. In particular, the graph contains no Hamiltonian circuit.

(f) *Give an example of a spanning tree for the graph, specifying the vertices and edges of the spanning tree.*

Vertices: a, b, c, d, e.

Edges $a\,c$, $b\,c$, $c\,d$ and $c\,e$.

(There are many other spanning trees.)

(g) *Given an example of an isomorphism $\varphi\colon V \to V$ from the given graph to itself which satisfies $\varphi(a) = d$. [You should specify the isomorphism as a function from the set $\{a, b, c, d, e, f\}$ to itself.]*

One such isomorphism maps vertices as follows: $\varphi(a) = d$, $\varphi(b) = d$, $\varphi(c) = c$, $\varphi(d) = a$, $\varphi(e) = b$.

(Any isomophism will map a vertex to some other vertex that has the same degree. Therefore any isomorphism from this graph to itself must map the vertex c to itself, since c is the only vertex of degree 4. The isomorphism must map the edge $a\,b$ to some other edge that does not involve the vertex c. Thus if $\varphi(a) = d$ then $\varphi(b) = e$. There are then two possibilities for constructing the required isomorphism. Either $\varphi(d) = a$, in which case $\varphi(e) = b$. Or else $\varphi(d) = b$, in which case $\varphi(e) = a$.)

4. (a) *Let graphs G_1 and G_2 be trees, where there are no vertices or edges that are common to both G_1 and G_2. Let v_1 be a vertex of G_1, let v_2 be a vertex of G_2, and let G be the graph consisting of the vertices and edges of G_1, the vertices and edges of G_2 and the edge $v_1 v_2$. Explain why the graph G is a tree.*

A tree is a connected graph with no circuits.

The graph G is connected. Indeed any two vertices of G_1 are joined by a walk, since G_1 is connected. Also any two vertices of G_2 are joined by a walk, since G_2 is connected. A vertex a of G_1 can be joined to a vertex b of G_2 by a walk obtained by concatenating a walk from a to v_1, the edge $v_1 v_2$ and a walk from v_2 to b. Thus any two vertices of G can be joined by a walk in G, and thus G is connected.

The graph G contains no circuit whose vertices are all contained in G_1, because the subgraph G_1 has no circuits. The graph G contains no circuit whose vertices are all contained in G_2, because the subgraph G_2 has no circuits. Thus any circuit in G would have to involve vertices from both G_1 and G_2. This circuit would have to traverse the edge $v_1 v_2$ in order to get from G_1 to G_2. But in order to return to the subgraph G_1 it would have to traverse this edge again in order to pass from G_2 to G_1. But this is impossible since, by definition, no circuit can traverse an edge of the graph more than once.

(b) *Let the graphs G, G_1 and G_2 be as in (a), let w_1 be a vertex of G_1 distinct from v_1, let w_2 be a vertex of G_2 distinct from v_2, and let G' be the graph formed from the graph G by adding an extra edge $w_1 w_2$. Is the graph G' a tree? [Justify your answer.]*

The graph G' is not a tree. The trees G_1 and G_2 are connected. Therefore one can construct a simple circuit by concatenating the following paths:

- the edge $v_1 v_2$, traversed from v_1 to v_2;
- a path in G_2 from v_2 to w_2;
- the edge $w_2 w_1$, traversed from w_2 to w_1;
- a path in G_1 from w_1 to v_1;