MA2C03: ASSIGNMENT 2
DUE BY FRIDAY, JANUARY 20
SOLUTION SET

BRIAN TYRRELL

Q1

(a) We obtain the following language, using formal expressions:

$$L = \{a, b\}^* \circ c$$

The grammar gives strings of the following format: rules (i) and (ii) give any configuration of a and b characters of any finite length (creating a string of length at least one). So rules (i) and (ii) generate $\{a, b\}^+$. Rule (iii) guarantees the string terminates with a c, so $\{a, b\}^+c$ is generated. Also note if we are given $< S >$, we can immediately apply rule (iii), in essence ignoring the a, b characters. So the language generated is precisely $\{a, b\}^*c$.

(b) Yes, it is a regular grammar. It's context-free and all of the production rules are in the required format.

(c) No, the grammar is not in normal form. We'll change the production rules as follows:
(i) $< S > \to a < S >$
(ii) $< S > \to b < S >$
(iii)' $< S > \to c < C >$
(iv)' $< C > \to \epsilon$
We have removed rule (iii) from the original grammar, added in rules (iii)' and (iv)', and added a new non terminal symbol $< C >$. These production rules still generate L - given $< S >$, we can generate $\{a, b\}^+$ still by (i) and (ii). Eventually we apply rule (iii)', and we can no longer generate any more a, b characters, as the only option for $< C >$ is rule (iv)'. This gives us $\{a, b\}^+c$. Note we can again 'skip' the a, b characters by applying rules (iii)' and (iv)' giving us c, i.e. $\{a, b\}^*c$ precisely is generated.

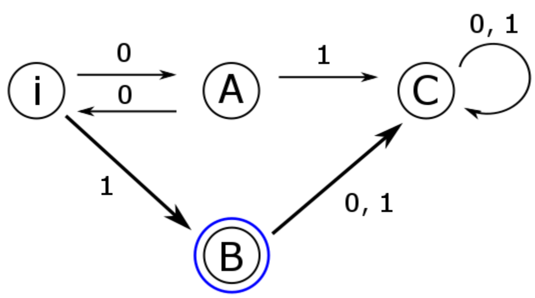
(d) Done in (a). The regular expression $(a^*b^*)^*c$ was also accepted, though in my opinion it's trickier and less clear than $\{a,b\}^*c$.

Common problems. Many people didn't seem to know what a regular expression is. Another common problem was, for (b), explaining the language satisfies the pumping lemma so it must be regular. THIS IS VERY WRONG - in the pumping lemma, we *assume* the language is regular and its most common application is to *disprove* regularity by contradiction.

Grading rubric. 5 for (a), 2 for (b), 8 for (c) and 5 for (d).

Q2

(a)



(b) Note that we can obtain the grammar by observing the FSA from (a). We simply write the steps taken to obtain a string in the language:

If $V \setminus A = \{<i>, <A>, <B>\}$, $A = \{0,1\}$ where $<i>$ is the start symbol, the production rules for this language are:

(1) $<i> \to 0 <A>$ - from $t(i,0) = A$.
(2) $<A> \to 0 <i>$ - from $t(A,0) = i$.
(3) $<i> \to 1 <B>$ - from $t(i,1) = B$.
(4) $<B> \to \epsilon$ - we don't want to leave the acceptance state.

We notice these rules generate L exactly, with repeated applications of (1) and (2) leading to an even number of zeros, ended by 1 by rules (3) and (4). Also note this grammar is regular, as FSA correspond to regular languages. Finally this is in normal form, as required.

There is a second way to approach this problem, from the ground up. Start with any grammar that generates the language, then try make it regular and normal.

Consider the following context free grammar (which we can make regular and normal if necessary):

$$V \setminus A = \{< S >\}$$

$$A = \{0, 1\}$$

P containing the following rules:

- $< S >\to 00 < S >$
- $< S >\to 1$

where $< S >$ is the start symbol.

This generates L; rules (1) and (2) give $0^{2m}1$ for all m (including 0). We can see this grammar isn't regular, however we can modify it so it becomes regular:

$$V \setminus A = \{< S >, < T >\}$$

$$A = \{0, 1\}$$

P containing the following rules:
(1) $< S >\to 0 < T >$
(2) $< T >\to 0 < S >$
(3) $< S >\to 1$
where $< S >$ is the start symbol.

...which is regular. This generates L, as any string in L is of the form $0^{2m}1$ and can be obtained by m uses of production rule (1) then (2), followed by an application of rule (3). These production rules don't generate any string not in L as well.

Finally, we can make this *normal* as follows:

$$V \setminus A = \{< S >, < T >, < U >\}$$

$$A = \{0, 1\}$$

P containing the following rules:
(1) $< S >\to 0 < T >$
(2) $< T >\to 0 < S >$
(3) $< S >\to 1 < U >$
(4) $< U >\to \epsilon$
where $< S >$ is the start symbol.

This is now normal and still generates L, as required.

Common problems. Not labelling all of the transitions was the biggest one. This is a major point - if you're writing a computer program to determine if a string is in a language, would you prefer the computer to return 'no' or to go into an infinite loop? Also, students seem very unsure what a grammar is and how to formulate production rules - we'll go over this in a later tutorial.

Grading rubric. 5 for both parts.

Q3

(a) Assume M is regular - it then has a pumping length p. We will now choose a string in terms of p that is particularly easy to analyse in the setting of the Pumping Lemma: consider the string $s = 0^p110^p$. By the pumping lemma, we can break s into three components; x, u, and y, with $u \neq \epsilon$ and $|xu| \leq p$.

Note that if $|xu| \leq p$, then xu must consist of 0's as the first p characters in s are 0's. Also if $u \neq \epsilon$, we must conclude $u = 0^k$ for some $k \geq 1$.

Therefore by the pumping lemma, for all n, $xu^ny \in M$, however by choosing $n = 2$ we obtain a string 0^q110^p where $q \neq p$. We reach a contradiction as this string cannot be in M, meaning M isn't regular, as required.

(b) Let

$$V = \{< S >\}$$
$$A = \{0, 1\}$$

and P be the following set of production rules:

- $< S > \to 0 < S > 0$
- $< S > \to 0110$

where $< S >$ is the start symbol. These rules generate M exactly.

Note that the including $< S > \to 11$ instead of the second rule won't work, as this would mean $11 \in M$, which is false.

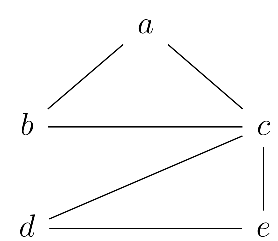
Common problems. There were too many similar answers in this one. True, there aren't a lot of ways to do (a) but the wording and structure of several student's work was too similar to be coincidental. You are allowed collaborate to reach an answer, but you MUST write up YOUR answers independently - that means a group is allowed solve a question together, however everyone must go off on their own and write their solution.

Mathematically, many students weren't general enough for (a). Choosing a string and showing it can't be pumped is an example, not a proof!

Grading rubric. 7 for (a) and 3 for (b).

Q4

(a)



(b) The incidence table:

	ab	bc	ac	ce	cd	de
a	1	0	1	0	0	0
b	1	1	0	0	0	0
c	0	1	1	1	1	0
d	0	0	0	0	1	1
e	0	0	0	1	0	1

The incidence matrix is thus:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

(c) The adjacency table:

	a	b	c	d	e
a	0	1	1	0	0
b	1	0	1	0	0
c	1	1	0	1	1
d	0	0	1	0	1
e	0	0	1	1	0

The adjacency matrix is thus:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

(d) The graph is not complete as a is not connected to d, for example.

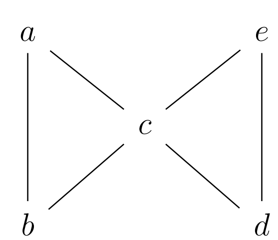
(e) The graph is not bipartite. If it was, we would be able to separate the vertices into two disjoint sets with no edges between the elements of each set. We see immediately if we were to look for two such sets of vertices, c must be in a set on its own as c connects to every other vertex. Therefore $V_1 = \{c\}$ and necessarily $V_2 = \{a, b, d, e\}$. However a is connected to b breaking the bipartite condition. Therefore this graph is not bipartite.

(f) This graph is not regular. If it was, then for some k every vertex is of degree k. However $\deg c = 4$ and $\deg d = 2$ meaning the graph cannot be k-regular for any k, so the graph is indeed not regular.

(g) Yes, the triangular subgraph $V = \{a, b, c\}$, $E = \{ab, ac, bc\}$ is 2-regular thus the graph does have a regular subgraph.

(h) The identity map φ satisfies $\varphi(c) = c$ and is an isomorphism of the graph.

(i) Viewing the graph like this makes things a little easier to see:



There is a vertical symmetry through c, meaning the map

$$\psi : \quad a \mapsto e \quad , \quad b \mapsto d$$

with $\psi(c) = c$ is also an isomorphism of the graph. Thus the identity isomorphism from (h) is not unique.

(There were multiple acceptable answers for this question.)

Note what we're being asked for here - **we are not asking for an example of a graph isomorph*ic* to (V, E), we are asked for an isomorph*ism* from (V, E) to itself.** Therefore for (h) and (i) we need to define a *map of vertices* that preserves the edge relation, and not just redraw (V, E).

Common problems. A shocking number of students got parts $(a), (b)$ and (c) wrong by misreading the question. People also have varying ideas to what a *complete* graph is - if you can't remember a definition/term, look it up *in the lecture notes*! Same goes for 'isomorphism'. Students lost a mark in (e) if they didn't prove/justify how the graph isn't bipartite.

Grading rubric. 2 for everything, except 4 for (i).