MA2C03 - DISCRETE MATHEMATICS - TUTORIAL NOTES

❄ **Brian Tyrrell** 🌲

12 & 13/12/2016

1 Regular Expressions & the Pumping Lemma

Regular expressions are the final way we will tackle regular languages.

Definition 1.1. Let A be an alphabet.

(1) $\emptyset$, ϵ and all elements of A are regular expressions.

(2) If w, w' are regular expressions, then so are $w \cup w'$, $w \circ w'$ and w^*.

Remark 1.1. There is an order of precedence when these operations are involved without explicit parentheses:

(1) $*$ (Kleene star)

(2) $\circ$ (concatenation)

(3) $\cup$ (union) $\Diamond$

Example 1.1. If $A = \{0, 1\}$ then

$$0^* \circ 1^* = \{0^m 1^n : m, n \in \mathbb{N}\}$$

$\Diamond$

We can show that a language is regular iff it is given by a regular expression. If we want to show that a particular language is generated by a particular regular expression, we can use the definition of *regular language* as follows:

Example 1.2. Let $A = \{0, 1\}$. The language $L = \{0^m 1^n : m, n \in \mathbb{N}\}$ is obtained from the sequence:

(1) $L_1 = \{0\}$

(2) $L_2 = L_1^* = \{0^m : m \in \mathbb{N}\} = 0^*$

(3) $L_3 = \{1\}$

(4) $L_4 = L_3^* = \{1^n : n \in \mathbb{N}\} = 1^*$

(5) $L = L_3 \circ L_4 = \{0^m 1^n : m, n \in \mathbb{N}\} = 0^* \circ 1^* = 0^* 1^*$ $\Diamond$

We know there are multiple equivalent formulations for regular languages, but given any language is there any way to tell if it's *not* regular without attempting to construct a regular expression, FSA, etc and deriving a contradiction?

The answer is yes: we can use the *Pumping Lemma*, which takes advantage of the fact that regular languages are infinite sets given by finite constructions, so must be repeating (or pumping) something:

Lemma 1.1. *If L is a regular language then there is a number p where if $w \in L$ is of length $\geq p$, then $w = xuy$ for $x, u, y \in L$ such that*

(1) $u \neq \epsilon$.

(2) $|xu| \leq p$.

(3) $\forall n \geq 0$, $xu^n y \in L$.

Proof idea:
Given a (sufficiently) long string xuy recognised by the FSA accepting L, the string must have passed through some state twice: say at x and xu the FSA is in state s. By pumping or repeating u, the FSA is constantly cycling back to state s - it's like the machine doesn't 'see' the pumping. So xuy, $xuuy$, $xuuuy$,...are all accepted, leading to the lemma. ■

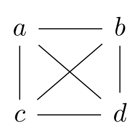
The applications of this are immediate, however we must be careful; this lemma can only be used to **disprove** a language is regular. An example of this:

Example 1.3. Show the language $L = \{ww : w \in \{0,1\}^*\}$ isn't regular.

Proof.
Assume L is regular; it then has a pumping length p. Consider the string $s = 0^p 1 0^p 1$ - i.e. $s = 00\ldots0100\ldots1$ with two sets of p zeros. We wish to divide s into x, u and y satisfying (1) and (2) of *Lemma 1.1*. As we need $|xu| \leq p$, our only option will be to have $u = 0^k$ for some $k \geq 1$. However now if we pump s, we end up with a string $0^q 1 0^p 1$ where $q \neq p$, meaning the pumped string isn't in L. This is a contradiction, meaning the language isn't regular, as required. ■

2 Graph theory

Example 2.1. Consider the following undirected graph: $V = \{a, b, c, d\}$ and $E = \{ab, ad, ac, bd, bc, cd\}$



This has an incidence table

	ab	ad	ac	bd	bc	cd
a	1	1	1	0	0	0
b	1	0	0	1	1	0
c	0	0	1	0	1	1
d	0	1	0	1	0	1

The incidence matrix follows suit. Similarly we can create an *adjacency* table, where vertices x and y are called adjacent if there is an edge between them.

Definition 2.1. A graph is called *complete* if $\forall v, w \in V$ with $v \neq w$, $\exists vw \in E$ - i.e. the graph is fully connected.
The complete graph with n vertices is denoted K_n.

Definition 2.2. A graph (V, E) is called *bipartite* if there exists subsets V_1, V_2 such that

(1) $V_1 \cup V_2 = V$.

(2) $V_1 \cap V_2 = \emptyset$.

(3) Every edge in E is of the form $v_1 v_2$ where $v_1 \in V_1$ and $v_2 \in V_2$

Similarly bipartite graphs have a notion of completeness, and a **complete** bipartite graph with $|V_1| = p$, $|V_2| = q$ is denoted $K_{p,q}$.

Example 2.2. Consider the following undirected graph: $V = \{a, b, c, d, e, f\}$ and $E = \{af, be, cd\}$. This has $V_1 = \{a, b, c\}$ and $V_2 = \{d, e, f\}$, and by examining E we see this forms a bipartite graph.