Automatic and asynchronously automatic groups were invented by J.W. Cannon, D.B.A. Epstein, D.F. Holt, M.S. Patterson and W.P. Thurston in \cite{CEHPT} a few years ago. The primary objective of this paper is to report, without proofs, on a number of new results about such groups. (Proofs will appear elsewhere, in due course.) Most of these results are concerned with the construction of new automatic and asynchronously automatic groups from old by means of amalgamated products. Our secondary objective here is to survey a little of the theory of automatic groups and the like, keeping the account reasonably self-contained.

2. Finite state automata

Let $A$ be the finite set $\{a_1, \ldots, a_q\}$ and let $A^*$ be the set of all words $w = a_1 \ldots a_n$ made up from letters in $A$ (including the empty word $e$); we term $n$ the length of $w$, which we denote by $\ell(w)$. This set $A^*$ together with the binary operation concatenation (i.e. juxtaposition) is a monoid, which we will make use of throughout. The subsets of $A^*$ are often referred to as languages over $A$. We shall be concerned with special languages over $A$ termed regular languages over $A$ or sometimes regular sets (over $A$). These depend for their definition, which will be given shortly, on the notion of a finite state automaton.

A finite state automaton is a quintuple $M = (S, Y, A, \tau, s_0)$, where

(i) $S$ is a finite set of states;
(ii) $Y$ is a subset of $S$, the set of accept states;
(iii) $A$ is a finite set, the alphabet, the elements of which are letters;
(iv) $\tau$ is a function from $S \times A \rightarrow S$, the transition function;
(v) $s_0$ is an element of $S$, the initial state or start state.
$M$ can be thought of as a machine with a head that scans a tape, which is in a horizontal position stretching from left to right. The tape is divided up into a finite number of squares. Each square has a letter printed on it. The left-most part of the tape is fed into $M$, which starts up in the initial state $s_0$. $M$ reads the first letter on the tape, and then moves on to the second letter and the machine goes into a new state. This new state is determined by the transition function $\tau$ and the letter that is being scanned by $M$. The process continues until the last letter on the tape is read. The machine then goes into a new state and stops. If the last state is an accept state, then the string of letters on the tape is accepted by $M$. Otherwise it is rejected. We define the language recognized by $M$ or the language accepted by $M$ or the language of $M$ to be the set $L(M)$ of words accepted by $M$. The language $L(M)$ of $M$ can therefore be described as follows. Given a word $w = a_1 \ldots a_n$, let $t_0 = s_0$ and let $t_i = \tau(t_{i-1}, a_i) \ (i = 1, \ldots , n)$. Then

$$L(M) = \{ w = a_1 \ldots a_n | t_n \in Y \}.$$

A language recognised by a finite state automaton is termed a regular language or a regular set.

The reader who is not familiar with these notions might find it instructive to think about the following example of a finite state automaton $M$. The set $S$ of states of $M$ consists of $s_0$, the initial state, $s_1, s_2, f$, the set $Y$ of accept states of $M$ consists of $s_0, s_1, s_2$, the alphabet $A$ of $M$ consists of $x, X$ and the transition function $\tau$ of $M$ is defined as follows:

$$\tau(s_0, x) = s_1, \tau(s_0, X) = s_2, \tau(s_1, x) = s_1, \tau(s_1, X) = f, \tau(s_2, x) = f, \tau(s_2, X) = s_2.$$

Then it is not hard to see that the language $L(M)$ of $M$ consists of all those words in $x$ and $X$ which do not contain a pair of consecutive letters of either the form $xX$ or $Xx$.

For more information on automata and languages, we refer the reader to the book by Hopcroft and Ullman [HU].

3. Padded alphabets

We shall be concerned with pairs of elements of $A^*$, i.e. with the direct product $A^* \times A^*$ of two copies of the monoid $A^*$. A little thought reveals