ON THE POWER OF THE COMPASS

(OR, WHY MAZES ARE EASIER TO SEARCH THAN GRAPHS)

by

M. Blum and W. Kozen

Memorandum No. UCB/ERL M78/64

6 September 1978

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ON THE POWER OF THE COMPASS
(or, Why Mazes are Easier to Search than Graphs)

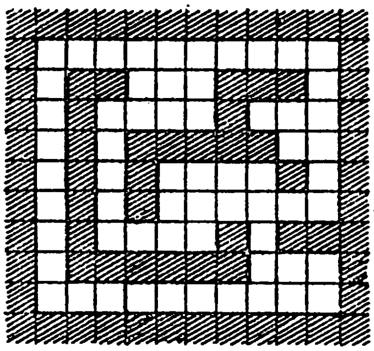
Manuel Blum†
Dexter Kozen‡

Department of Electrical Engineering and Computer Sciences
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

Appeared in: Proceedings of the 19th Annual Symposium on Foundations of Computer Science, Oct. 1978.

1. Introduction

A *maze* is a finite, two-dimensional, obstructed checkerboard.



To search a maze, a finite automaton, started on any cell, must eventually visit every reachable cell without passing through any barriers. In each step, the automaton determines which of its neighbor cells are reachable in one step and then, depending on its state, moves north, east, south, or west one cell.

In 1974 Lothar Budach gave a very long (175 manuscript pages) but readable proof that no single automaton can search all mazes. The proof was quite formal, making nontrivial use of concepts and techniques of category theory.

About the same time, A.N. Shah gave a finite automaton with 5 pebbles which could search an arbitrary maze (the automaton may drop a pebble on a cell it is visiting, then upon returning to that cell later on can sense the pebble's presence, and if desired pick it up and move it to a new cell). Shah also conjectured that fewer than 5 pebbles would not suffice. The first of our two main results is that, contrary to Shah's conjecture, the search can be implemented with just two pebbles. The question is still open whether a finite automaton with just one pebble can search any maze.

The algorithm given has two parts, represented by the following two lemmas:

Lemma 1. A single automaton without pebbles can search any maze, provided the westernmost of all southernmost squares of any barrier is marked.

This lemma is proved by showing how the marked squares of the barriers delineate a spanning tree covering the maze.

Lemma 2 (Unique Point Lemma). A one-counter automaton, visiting cell P on the boundary of a barrier, may search the boundary of the barrier and return to P and stop, with the knowledge whether or not P is the westernmost of all southernmost points of that barrier. Moreover, the counter never need hold a number larger than the perimeter of the barrier.

Our proof technique also yields the following related results:

(1) there is a two pebble automaton that can search all mazes (the two pebbles simulate the counter);

(2) there are two automata which together can search all mazes;

(3) there is a logspace algorithm to search mazes, a vast improvement over the naive linear space algorithm which constructs a map of the maze.

Mazes and regular planar graphs appear similar on the surface, but in fact they differ substantially. The primary difference is that an automaton in a maze has a *compass:* it can distinguish N,E,S,W. A compass can provide the automaton with valuable information, as shown by the second of our two main results, namely that no three automata together can search all finite planar cubic graphs.

These two results taken together say that mazes are strictly easier to search than regular planar graphs, answering a question of Blum and Sakoda (1977).

The proof that no single automaton can search all finite planar cubic graphs is quite straightforward, in contrast to Budach's proof of the corresponding result for mazes. The proof that no pair of automata can search all finite planar cubic graphs is an order of magnitude harder, but once the tricks are established the proof is routine.

The proof that no three automata can search all such graphs is again an order of magnitude harder than the proof for two. We believe that a formal treatment à la Budach would yield 175 pages, as well. The proof makes use of concepts of differential geometry. In particular it requires the development of the geometry of the *regular tessellations* {p,q} (notation is from Coxeter (1963)). The graph {p,q} is the unique planar graph of degree q, all of whose faces have p edges. The geometry of the five finite {p,q}, i.e. those for which

$$\frac{1}{p}+\frac{1}{q}>\frac{1}{2}$$

corresponds to spherical geometry. The geometry of {3,6}, {4,4}, and {6,3}, i.e. those {p,q} for which

$$\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$$

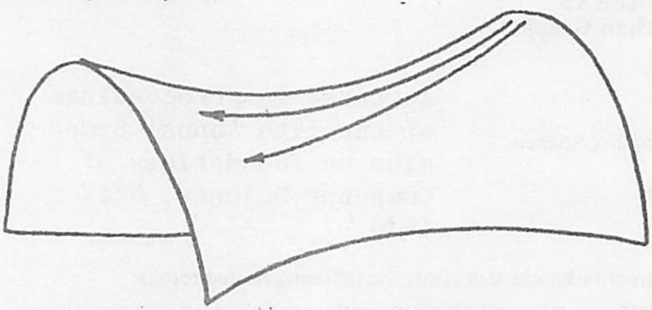
corresponds to Euclidean plane geometry. The geometry of the remaining {p,q}, i.e. those for which

$$\frac{1}{p}+\frac{1}{q}<\frac{1}{2}$$

corresponds to hyperbolic geometry, first developed by Lobachevsky and Bolyai and later refined by Gauss. Paths of an automaton in {p,q} are periodic, due to the rich automorphism structure of {p,q}; they correspond to curves of constant geodesic curvature in the surface corresponding to {p,q}.

†Supported by NSF grant MCS75-21760-A01.
‡Supported by NSF grant MCS74-07636-A01. Present address IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

A key step in the proof is that, for suitable p and q, any two initially parallel paths of the automaton in {p,q} diverge, in much the same way that initially parallel geodesics diverge in a hyperbolic plane or pseudosphere.



We develop the geometric tools in a formal way as far as possible, since they may be of independent interest. In particular we formalize the concept of Gaussian curvature for the regular tessellations, and prove a discrete analog of the Gauss-Bonnet Formula, one of the central tools of differential geometry.

Our lower bound proof for three automata does not appear to generalize, and in fact the problem remains open whether there is a finite set of automata which together can search all finite planar cubic graphs.

2. Preliminary definitions and notation

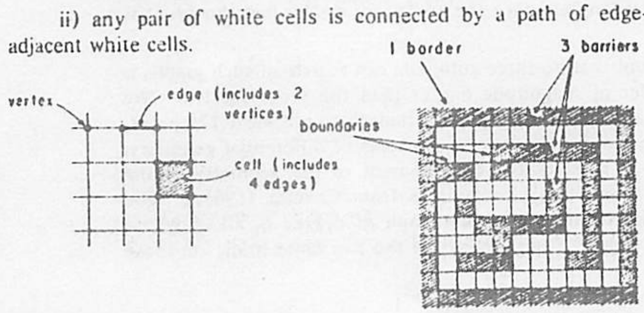
Z, N, and R denote the integers, nonnegative integers, and real numbers respectively.

Mazes

A *vertex* is a point in the Euclidean plane with integer coordinates. An *edge* is a unit-length segment connecting two vertices. A *cell* is a region of unit area enclosed by four edges. A *maze* consists of an assignment of either black or white to each cell, so that

i) there are only finitely many white cells,

ii) any pair of white cells is connected by a path of edge-adjacent white cells.



A *barrier* is any maximal connected set of black cells. The unique infinite barrier is called the *border*. Since any two white cells are connected by a path of edge-adjacent white cells, it follows that barriers are simply connected, i.e. contain no holes. The maze illustrated above contains three barriers, including the border.

The *boundary* of a barrier is the set of edges that separate black cells from white cells. A white cell edge-adjacent to a black cell is called a *boundary cell.*

Automata in mazes

A *finite automaton* consists of a finite control with start and halt states and a transition function. A finite set of automata search a maze as follows. The automata are started together on a white cell, all in their start states. In one step, each automaton determines which other automata are visiting the same cell and their current states, which adjacent cells are white, and its own current state. Based on this information, it moves to an edge-adjacent white cell and enters a new state.

We may equip an automaton with a finite number of *pebbles*, each with a unique name. At the start, the automaton is carrying all of its pebbles. Thereafter, it is always carrying some subset of its pebbles, and the rest are lying on white cells of the maze. In each step the automaton determines the names of the pebbles it is carrying and the names of the pebbles lying on the cell it is visiting. It may use this information to help determine its next transition. In each step the automaton may pick up pebbles from the cell it is visiting or deposit some.

We may also equip an automaton with one or more *counters*. A counter holds a nonnegative integer, initially zero. In each step, the automaton can increment or decrement the counter by one and test for zero.

A set of automata is said to be capable of searching a maze if every reachable white cell is visited by some automaton eventually. The automata need not halt.

For a more formal treatment, the reader is referred to Budach (1977).

Embedded graphs

Mazes are a special case of the following more general construct, which we will use in sections 4, 5, and 6.

An *embedded graph* G is an undirected, connected, planar graph equipped with an embedding in the plane. VG represents the (possibly infinite) set of vertices of G, EG its undirected edges, and FG its faces. G^* is the planar dual of G. $\vec{E}G$ is the set of directed edges formed from edges in EG,

$$\vec{E}G = \{(u,v) \mid \{u,v\} \in EG\}.$$

The embedding of G imparts an orientation to the edges incident to a vertex. By an *automorphism* of G we will always mean an automorphism preserving this orientation. The group of all such automorphisms is denoted Aut(G). A *subgraph* H of an embedded graph G is a full subgraph of G (i.e. one for which $\{u,v\} \in EH$ whenever $u,v \in VH$ and $\{u,v\} \in EG$) equipped with the embedding inherited from G.

A class of embedded graphs we will find particularly useful is the class of *regular tesselations* {p,q}, where {p,q} is the unique embedded graph of degree q, all of whose faces have p edges. Note that $\{p,q\}^* = \{q,p\}$. The notation {p,q} is from Coxeter (1963).

A *maze* may then be viewed as a finite connected subgraph of {4,4}.

Automata in embedded graphs

Finite automata in embedded graphs are defined with respect to an arbitrary but fixed $k \in N$. They may search only embedded graphs which are regular of degree k or their subgraphs. Suppose H is a subgraph of regular embedded graph G. A collection of finite automata searches H by visiting its edges. In one step, an automaton visiting (directed) edge (u,v) determines which other automata are visiting (undirected) edge {u,v} and their current states, and which edges of G incident to v are present in H. Based on this information, it moves to one of these edges (v,w). The automata run synchronously. Automata in mazes represent the case $k = 4$.

3. Algorithms for searching mazes

In this section we will prove

Theorem 3.1. There is a finite automaton with one counter which can search any maze and halt.

This yields the following related results:

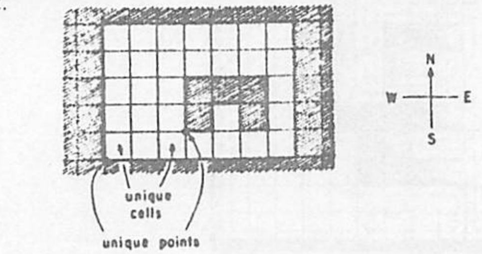
Corollary 3.2. There is a finite automaton with two pebbles which can search any maze and halt.

Corollary 3.3. There are two finite automata which together can search any maze and halt.

The 1 counter or 2 pebbles will be used only to measure y-distance (latitude) along a boundary.

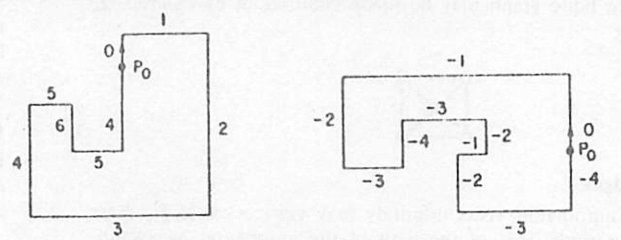
Definition. The *unique point* of a boundary, BDRY, (or its associated barrier) is the unique point $(x_0,y_0) \in$ BDRY such that for all $(x,y) \in$ BDRY$[y_0 \leq y$ or $(y_0 = y$ and $x_0 \leq x)]$.

The *unique cell* of a boundary (or its associated barrier) is the unique white cell whose NE or SW vertex is the boundary's unique point.



Unique Point Lemma. There exist both 2 pebble and 1 counter automata that one may place on any white cell, C, of a maze together with 2 pebbles or an empty counter (whichever the case may be), in state q_{NE} (or q_{SW}). The automaton, after some moving about in the maze, will *return to* C with its 2 pebbles or empty counter and *stop* in state q_{NE}^{YES} (q_{SW}^{YES}) if the NE (SW) vertex of C is a "unique point" of a boundary and in q_{NE}^{NO} (q_{SW}^{NO}) if not.

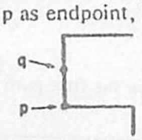
Let C denote a closed simple curve in 2-dimensional Euclidean space composed entirely of a finite number of horizontal and vertical line segments. Let p_0 be a point in the interior of some vertical line segment. For p in the interior of some line segment of C, let
$v(p)$ = [total number of right hand turns − total number of left hand turns that must be made in a walk from p_0 to p that begins at p_0, initially goes north a nonzero distance, and continues in the same direction along C until p is reached].



The ±4 Lemma. $v(p_0) = 4$ if the walk from p_0 to p_0 is homotopic to a clockwise loop. $v(p_0) = -4$ if the walk is homotopic to a counterclockwise loop.

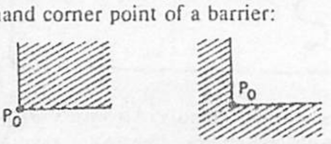
Mod 3 Corollary. $v(p_0) = 1 \bmod 3$ if the walk from p_0 to p_0 is homotopic to a clockwise loop. $v(p_0) = 2 \bmod 3$ if the walk from p_0 to p_0 is homotopic to a counterclockwise loop.

Definition. For p = a corner point and q = a point on a vertical line segment having p as endpoint, let $v(p) = v(q)$.

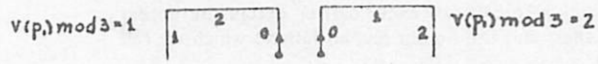


Algorithm and informal proof of the Unique Point Lemma for 1-counter automata

A 1-counter automaton can check if a boundary point, p_0, is the unique point of that boundary as follows. First it checks that p_0 is a lower left-hand corner point of a barrier:



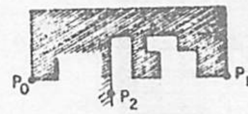
Then it travels north from p_0 along the boundary using the counter to keep track of vertical distance or *latitude*. The counter interrupts the automaton each time it passes a point having the same latitude as p_0. While moving along the boundary, the automaton uses its internal states to keep track of $v(p) \bmod 3$. When the automaton is first interrupted at a point p_1 (at the same latitude as p_0), it knows whether p_1 lies west or east of p_0, depending on whether $v(p_1)$ mod 3 = 1 or 2, respectively (this follows from the mod 3 corollary).



Case 1: p_1 lies west of p_0. The automaton retraces its path back to p_0 and halts there. In this case, p_0 is *not* the unique boundary point.

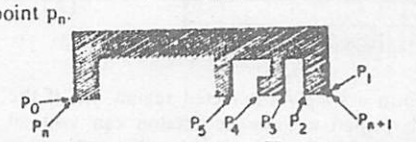
Case 2: p_1 lies east of p_0. In this case, the automaton continues moving along the boundary (as before) until one of the following two conditions occurs:

Condition 1. A boundary point p_2 is found at a latitude below p_0.



In this case, the automaton can retrace this path back to p_0 and halt. p_0 is *not* the unique point of that boundary.

Condition 2. The automaton discovers a finite sequence of successive points $p_2,p_3,...$ in the same latitude as p_1 with the property that each point p_{i+1} $(i \geq 1)$ lies west of the preceding point p_i until for the first time a point p_{n+1} is found that lies east of the preceding point p_n.



In this case, $p_n = p_0$ and $p_{n+1} = p_1$. The automaton then retraces its path back from p_{n+1} to p_n and halts back where it started. In this case, p_0 *is* the unique point of the boundary.

Finally, the automaton can easily decide whether or not p_0 lies on the (outside) border on the basis of whether the cell northeast or southeast of p_0 is black.

Replacement of the counter by 2 pebbles

To replace the counter by 2 pebbles, note that the automaton uses the counter only to measure y-displacement while traveling along the boundary of a barrier. An automaton can as easily use 2 pebbles to store the count, with the distance along the boundary between the 2 pebbles serving as counter contents and the lead pebble serving to mark the position of the original 1-counter automaton.

The single counter automaton may also be replaced by two finite automata without counters or pebbles. This construction we leave to the reader.

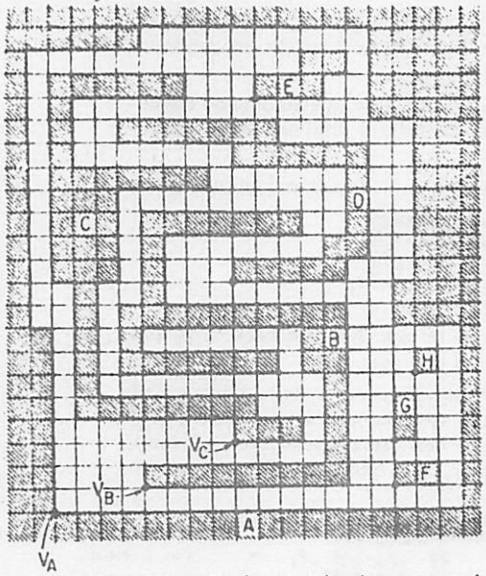
Proof of Theorem 3.1.

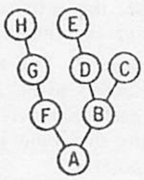
Let us assign to every vertex of a maze a color. Unique points are to be colored green and all other vertices are to be colored white.

Extend the definition of a finite automaton so that in any cell and in any state there, the automaton can determine which, if any, of the cell's 4 vertices are green. As usual, the automaton can determine which of its neighboring cells are white and then, based on this information, can either halt or cross an edge into a neighboring white cell and change its state appropriately. This we call a *finite green-eyed automaton.*

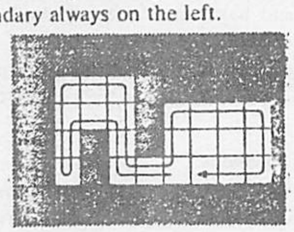
A 1-counter or 2-pebble automaton is at *least* as powerful as a finite green-eyed automaton (with no pebbles and no counter) in a maze whose unique points are colored green. This follows from the unique point lemma. Thus we may (and shall) complete this proof by showing how a finite green-eyed automaton can search a maze in which unique points are colored green.

We begin by defining an ordering on the barriers of a maze. Let X and Y be barriers. Let V_X and V_Y denote their unique points. Say that X is *father* of Y iff Y is not the border and X is the first barrier reached by moving due south from V_Y. This ordering of the barriers forms a tree (since every barrier except the border has exactly one father, and the border has no father) which we call the *tree of barriers:/*

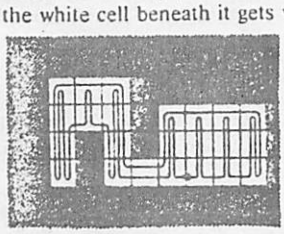




If the white cells form a simply connected region, i.e. if the only barrier is the border, then a finite automaton can visit all (white) *boundary cells* by moving from one such cell to the next, keeping the boundary always on the left.

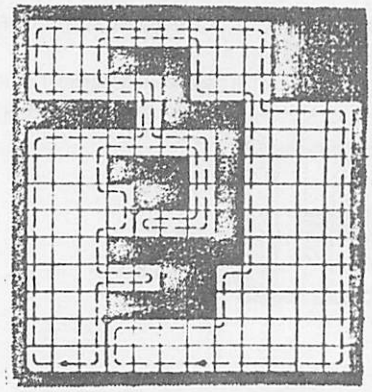


To search an entire simply connected maze, modify the above procedure so that each time the automaton steps from one boundary cell to the next, it first goes into a subroutine that causes the automaton to move north until it reaches a barrier and then to return south whence it came before going on to the next step. This way, each white cell interior to a simply connected maze gets visited immediately after the white cell beneath it gets visited.



Any maze can be converted to a simply connected one by relabeling all vertical edges that lie between each green vertex and the barrier immediately beneath it as boundary edges. The figure below shows all boundary edges plus all those specially labelled as such drawn heavily in black. The path of an automaton along this boundary is shown dotted.

Finally, in order to halt, the finite green-eyed automaton has only to check that it twice visited the (only) border cell having a green vertex.



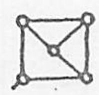
4. Lower bounds for 1 and 2 automata in planar graphs

This and the next two sections will be devoted to proving that no three finite automata together can search all finite cubic embedded graphs. In this section we prove such a lower bound for 1 and 2 automata.

In contrast to Budach's corresponding result for mazes, the following theorem is easy to prove:

Theorem 4.1. No single automaton can search all finite cubic embedded graphs.

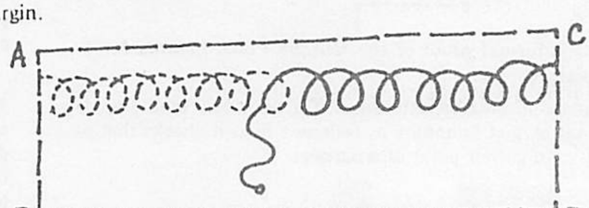
Proof. Start the automaton on {6,3}. Since all vertices are of degree 3 and no other automata are present, it eventually enters a cycle of states, and thereafter its path is periodic. If this periodic path describes a cycle of {6,3}, then cut away all of {6,3} except the finite portion ever visited by the automaton, plus a little margin. The resulting finite graph may be made cubic again by *cauterizing*, i.e. attaching



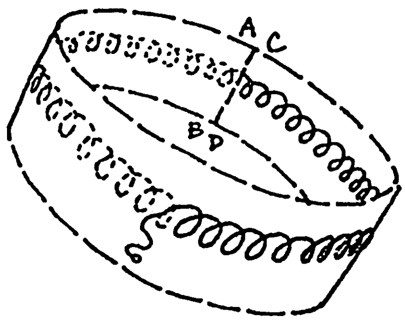
to the cut edges.

If the automaton visits infinitely may vertices of {6,3}, then extend the periodic part of the path of the automaton backwards and forwards several periods.

Cut away all of {6,3} except the vertices on this path plus a slight margin.

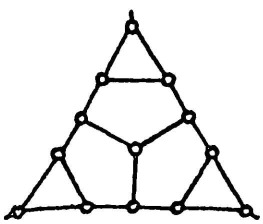


Twist the resulting graph around and identify AB with CD so that the ends of the periodic path match up, forming a continuous periodic cycle. Cauterize cut edges as above.



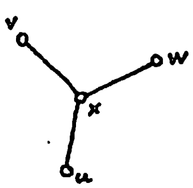
In both cases we have constructed a finite embedded graph such that the automaton moves in a cycle without searching all of the graph. This completes the proof.

The proof that no *two* automata can search all finite cubic embedded graphs is somewhat harder. The main idea is to construct *hypernodes* -- finite, rotationally symmetric, embedded graphs which are cubic except for three exterior vertices, e.g.



All nodes of a cubic graph G can be replaced by hypernodes ϕ to get $G \circ \phi$. We then observe an automaton's motion in $G \circ \phi$, viewing occurrences of ϕ as black boxes, ignoring the automaton's motion inside ϕ. The key lemma (4.5) states that for any automaton M, a special hypernode ϕ_M may be constructed so that M's motion in $G \circ \phi_M$ is *degenerate*, i.e. between copies of ϕ_M always goes left, always goes right, or always backs up.

Let us restrict our attention to the behavior of an automaton M in cubic embedded graphs. For such graphs, the orientation defined by the embedding takes the form of unique left and right turn functions $L,R:\vec{E}G \rightarrow \vec{E}G$, as illustrated. We also include $B:\vec{E}G \rightarrow \vec{E}G$ for backing up.



$L(u,x) = (x,v)$
$R(u,x) = (x,w)$
$B(u,x) = (x,u)$

The functions L, R, and B are one-one and onto.

In the absence of other automata, the sequence of states that M assumes and its turning behavior are independent of the graph it is searching and quickly become periodic. In order to concentrate on this periodic behavior, we will isolate the *cycles* of M, i.e. those states and transitions of M which remain after removing transitions involving input from other automata and non-cycle states (p is a *cycle state* if, when started in p, M reenters p infinitely often). The cycles of M are governed by two functions

NEXTSTATE:Q→Q
DIRECTION:Q→{L,R,B}

where Q is the set of cycle states When visiting edge (u,v) in state $p \epsilon Q$ with no other automata present, M moves into state NEXTSTATE(p) and to edge DIRECTION(p)(u,v). NEXTSTATE is one-one and onto Q.

If M is a set of cycles and G is a cubic embedded graph, the set of *configurations* for M and G is the set

$$\text{CONFIG} = \vec{E}G \times Q$$

describing a possible current position of M in G and current state. The function NEXTCONFIG:CONFIG→CONFIG describes the motion of M in one step:

NEXTCONFIG(u,v,p)=(DIRECTION(p)(u,v),NEXTSTATE(p)).

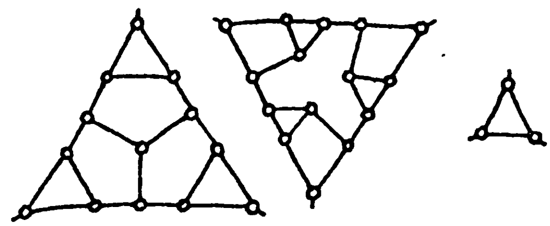
Since L, R, B and NEXTSTATE are 1-1 and onto, so is NEXTCONFIG.

A *hypernode* ϕ is a finite embedded graph with three distinguished vertices such that

i) all vertices are of degree 3 except the 3 distinguished vertices, which are of degree 2;

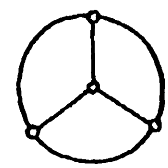
ii) the 3 distinguished vertices occur on the boundary of the exterior face of ϕ (determined by the embedding);

iii) ϕ has a nontrivial automorphism.

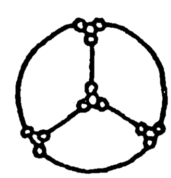


We also allow the *trivial hypernode*, consisting of a single vertex and no edges.

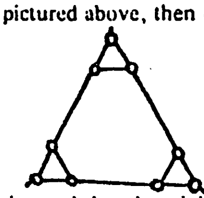
Given a cubic embedded graph G and hypernode ϕ, we construct embedded graph $G \circ \phi$ by replacing all nodes of G with ϕ. The embedding of $G \circ \phi$ is inherited from G and ϕ. For example, if G is the graph



and ϕ is the small hypernode pictured above, then $G \circ \phi$ =



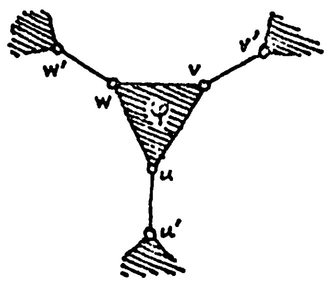
If ϕ, ψ are hypernodes, then so is $\phi \circ \psi$. For example, if ϕ is the small hypernode pictured above, then $\phi \circ \phi$ is the hypernode



Note that $\circ$ is associative and that the trivial hypernode serves as an identity for $\circ$.

To be completely precise, we must associate with $G \circ \phi$ a morphism $h:G \circ \phi \rightarrow G$ which collapses ϕ, so that we know which occurrences of ϕ in $G \circ \phi$ replaced vertices of G. We will take this for granted, using "occurrence of ϕ in $G \circ \phi$" for "inverse image of a vertex of G under h". A *hyperedge* of $G \circ \phi$ is the inverse image of an edge of G under h, and a *hyperface* is the inverse image of a face of G under h. We denote by $Aut_\phi(G \circ \phi)$ the group of automorphisms of $G \circ \phi$ preserving occurrences of ϕ. Then $Aut_\phi(G \circ \phi) \cong Aut(G)$.

Suppose ϕ occurs in $G \circ \phi$ as illustrated.



The embedding of ϕ imparts an orientation to the hyperedges incident to u, v, and w. This orientation is given by left and right turn functions L_ϕ, R_ϕ and backup function B_ϕ.

$$L_\phi(u',u)=(w,w')$$

$$R_\phi(u',u)=(v,v')$$

$$B_\phi(u',u)=(u,u').$$

L_ϕ, R_ϕ, and B_ϕ are one-one and onto the set of hyperedges.

Let M be an automaton consisting only of cycles,

$$M = \langle Q, NEXTSTATE, DIRECTION \rangle .$$

Suppose M is started in state p visiting hyperedge (u',u). In the next step it enters ϕ. At some future time it must emerge from ϕ, since NEXTCONFIG is one-one. If it emerges in state q, we take

$$NEXTSTATE_\phi(p)=q$$

and

$$DIRECTION_\phi(p)=L \quad \text{(respectively R, B)}$$

if it emerges on hyperedge $L_\phi(u',u)$ (respectively $R_\phi(u',u)$, $B_\phi(u',u)$). It follows from the fact that NEXTCONFIG is one-one that $NEXTSTATE_\phi$ is one-one and onto Q, thus partitions Q into disjoint cycles. Moreover, if p is a state of cycle C of M, then $NEXTSTATE_\phi(p)\epsilon C$, so this partitioning refines the partitioning of NEXTSTATE.

In this way we have defined a new automaton

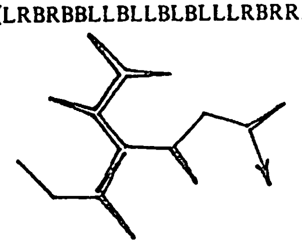
$$M_\phi = \langle Q, NEXTSTATE_\phi, DIRECTION_\phi \rangle$$

such that the behavior of M_ϕ in G mimics the behavior of M in $G\circ\phi$. We could formalize this statement with a commutative diagram involving paths of M in $G\circ\phi$, paths of M_ϕ in G, and the collapsing morphism $G\circ\phi \to G$, but for now we trust in the reader's intuition and just state

Lemma 4.2. The behavior of M_ϕ in G mimics the behavior of M in $G\circ\phi$.

Let C be a cycle of automaton M. C is an automaton itself. There is a two-way infinite periodic sequence $x_C \epsilon \{L,R,B\}^Z$ describing the turning motion of M when it is in cycle C. The *reduced sequence* $\underline{x}_C$ is x_C modulo trees. For example, if

$$x_C = (LRBRBBLLBLLBLBLLLRBRRLBL)^Z$$



then

$$\underline{x}_C = (LLRLRR)^Z.$$

For any periodic x, either $\underline{x}\epsilon B^Z$ or $\underline{x}\epsilon\{L,R\}^Z$. We say x is *degenerate* if $\underline{x}\epsilon\{L^Z,R^Z,B^Z\}$. A cycle C is *degenerate* if x_C is. An automaton M is *degenerate* if all its cycles are. For example, if $\underline{x}_C = L^Z$, and if M is in cycle C on {6,3}, then M is going counterclockwise around some face, with possible tree-like excursions away from that face at various points, but always returning to that face.

We now show how to construct a hypernode ϕ_M to make M degenerate.

Lemma 4.3. For any nondegenerate cycle C, there is a hypernode ϕ_C such that some cycle of C_{ϕ_C} is degenerate.

Proof. We may assume without loss of generality that x_C is reduced modulo trees. Since C is nondegenerate,

$$x_C \epsilon \{L,R\}^Z - \{L^Z,R^Z\}.$$

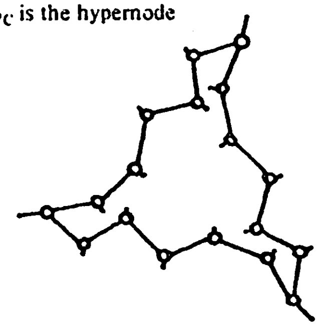
There are three cases:

i) x_C contains 2 consecutive L's.

ii) x_C contains 2 consecutive R's.

iii) $x_C = (LR)^Z$.

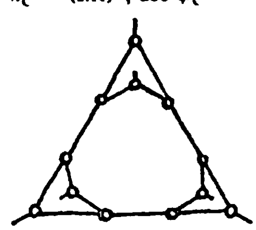
Case i). Suppose $x_C = (LyL)^Z$ for some $y \epsilon \{L,R\}^*$. Construct ϕ_C from 3 copies of y, placed end to end, oriented clockwise. Free edges are cauterized. For example, if $x_C = (LyL)^Z$ with y = LLRLR, then ϕ_C is the hypernode



If p is a state of C such that p occurs at the start of period LyL of x_C, and if D is the cycle of C_{ϕ_C} containing p, then $x_D = L^Z$.

Case ii) is analogous.

Case iii). If $x_C = (LR)^Z$, use ϕ_C =



Then some cycle D of C_{ϕ_C} has $x_D = B^Z$.

Lemma 4.4. If C is degenerate and ϕ is any hypernode, then all cycles of C_ϕ are degenerate.

Proof. Follows from Lemma 4.2.

Lemma 4.5. For any M, there is a hypernode ϕ_M such that M_{ϕ_M} is degenerate.

Proof. Execute the following program:

```
begin N := M;
    φ := the trivial hypernode;
    while N has a nondegenerate cycle C do
        begin φ := φC∘φ;
              N := Nφc
        end
end
```

The program always halts, since each iteration of the while loop strictly increases the number of degenerate cycles, by Lemmas 4.3 and 4.4, and there can be at most $|Q|$. To show correctness, consider the statement

$$N = M_{\phi}.$$

This statement is certainly true upon first entry to the while loop. Moreover, it is an invariant of the loop, since if $N = M_{\phi}$ then

$$N_{\phi_C} = (M_{\phi})_{\phi_C} = M_{\phi_C \circ \phi}.$$

Thus $N = M_{\phi}$ upon termination. But since N consists only of degenerate cycles upon termination, the final value of ϕ is the desired hypernode ϕ_M.

Theorem 4.6. No two finite automata together can search all finite cubic embedded graphs.

Proof. Let M', N' be the automata consisting of all cycles of M, N respectively. Let $\phi = \phi_{M'} \circ \phi_{N'}$. By Lemmas 4.4 and 4.5, M'_{ϕ} and N'_{ϕ} are degenerate.

Start M and N together in $\{6,3\}\circ\phi$. If M and N see each other only finitely often, then M and N eventually revert to M' and N'. But by 4.2, M' and N' in $\{6,3\}\circ\phi$ are simulated by M'_{ϕ} and N'_{ϕ} in $\{6,3\}$, which are degenerate; thus M and N thereafter stay confined within a bounded region of $\{6,3\}\circ\phi$, tracing closed loops. The argument of Theorem 4.1 now applies.

Otherwise, M and N see each other infinitely often. Eventually a configuration is repeated, and due to the rich automorphism structure of $\{6,3\}\circ\phi$, the behavior of M and N is periodic thereafter. The argument of Theorem 4.1 now applies. This completes the proof.

5. The geometry of {p,q}

In this section we develop the geometric tools necessary to the lower bound for three automata to be proved in section 6. This section may be skipped on first reading. Proofs will be sketched or omitted. We will be concerned with the geometric properties of the regular tessellations {p,q} defined in section 2, particularly {3,q} for $q > 6$.

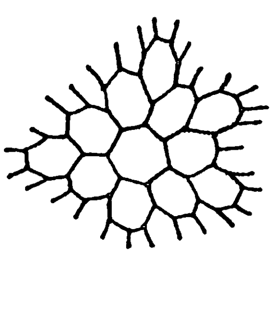
The regular tesselation {p,q} is called *spherical, plane,* or *hyperbolic,* if

$$\frac{1}{p}+\frac{1}{q} \;\; >\frac{1}{2}, \;\; =\frac{1}{2}, \text{ or } \;\; <\frac{1}{2},$$

respectively.

In studying the regular tesellations it is helpful to associate with each {p,q} a surface of constant Gaussian curvature. With a spherical (resp. plane, hyperbolic) tessellation we associate a sphere (resp. Euclidean plane, hyperbolic plane), a surface of constant positive (resp. zero, negative) Gaussian curvature. The geometry of {p,q} is the discrete analog of the geometry of its associated surface. This correspondence has been observed by Coxeter (1963).

The five spherical tesselations {3,3}, {4,3}, {3,4}, {5,3}, {3,5} are the only finite ones; they are the *tetrahedron, cube, octahedron, dodecahedron,* and *icosahedron,* respectively. There are only three plane tesselations: {6,3}, {4,4}, {3,6}. The remaining tesselations are the hyperbolic tesselations. A portion of {7,3} is illustrated.



The groups Aut({p,q}), denoted $[p,q]^+$ by Coxeter (1963), have been studied by Miller (1902), Brahana (1928) and others (see Coxeter and Moser (1972)).

Elements of Aut({p,q}) are denoted σ, τ. Every σ is either a *rotation* (if it has finite order) or a *translation* (if it has infinite order). *Reflections* are not allowed, since they do not preserve orientation. σ is a rotation iff it preserves an edge, face, or vertex, in which case it has order 2, divisor of p, or divisor of q, respectively. {p,q} is *symmetric* in the sense that for every pair of edges (u,v), (u',v'), there is a unique σ with $\sigma(u) = u'$ and $\sigma(v) = v'$.

d denotes the shortest-path metric on V{p,q}. If A, B are sets of vertices, we take

$$d(A,B) = \inf_{u \in A} \inf_{v \in B} d(u,v),$$

although d is no longer a metric when extended to sets. Any automorphism is an isometry with respect to d.

Curves

Let {p,q} be an infinite tesselation. A *curve* in {p,q} is a two-way infinite path in {p,q} such that any 3 consecutive vertices are distinct. Curves may be directed or undirected. A *parametrization* of a curve is a map $x: Z \rightarrow V\{p,q\}$ such that

i) x(m), x(m+1) are adjacent

ii) $x(m-1) \neq x(m+1)$ for any m.

x, y, z denote parametrizations. x and y parametrize the same directed curve if there is a c such that $x(m) = y(c+m)$ for all m. x and y parametrize the same undirected curve if for some c, either $x(m) = y(c+m)$ for all m or $x(m) = y(c-m)$ for all m. We may view a curve as the equivalence class of all its parametrizations. We will usually fix a parametrization, but most of the properties of curves to be studied are independent of the choice of parametrization.

Vx denotes the vertices of x and Ex its (undirected) edges. A curve x is *finite* if Vx is. A curve y is a *subcurve* of x if $Ey \subseteq Ex$. We write d(x,y) for d(Vx,Vy). x(m,n) denotes the *finite segment* of x consisting of x(m),x(m+1),...,x(n). We also allow *semi-infinite segments* by letting $m = -\infty$ or $n = \infty$. s,t denote finite segments of curves. s is a *loop* if its endpoints coincide. We define

$$|x(m,n)| = n-m,$$

$$||x(m,n)|| = d(x(m),x(n)).$$

s is a *shortest path* if $||s|| = |s|$. x is a *shortest path* if all its finite segments are.

Simplicity

A curve is *simple* if the relation

$$\{ <x(m),x(m+1)> \mid m \in Z \}$$

is a one-one function. For infinite x, this means no multiple vertices. For finite x, this means x traces a simple circuit in {p,q}. A *simple loop* of x is a loop with no multiple vertices except its endpoints. It is a consequence of the Jordan curve theorem that any simple curve partitions {p,q} into exactly 2 (plane-) connected regions. If x is directed, these regions may be designated *left* and *right* in the obvious way. A finite simple curve is *positively oriented* if it bounds the region to its left.

Periodicity

If $c \in Z$ then c is a *period* of x if there is a σ in Aut({p,q}) with $\sigma(x(m)) = x(c+m)$ for all m. For each c, if such a σ exists then it is unique, and is denoted $\sigma_{x,c}$. The set

$$Aut(x) = \{\sigma_{x,c} \mid c \text{ is a period of } x\}$$

is a subgroup of Aut({p,q}) isomorphic to Z or some cyclic group Z_{n}. x is said to be *periodic* if it has a nontrivial period.

Proposition 5.1. If x is periodic then the following are equivalent:

i) x is infinite

ii) $Aut(x) \simeq Z$

iii) $\sup \{ |m-n| \mid d(x(m),x(n)) < b \} < \infty$.

In the sequel, a *period* of x, otherwise unqualified, will mean a nonzero period of x.

Adherence, divergence, approximation

Two curves are *adherent* if they stay close together. More precisely, for $A,B \subseteq V\{p,q\}$, we say A *adheres to* B if

$$\sup_{u \in A} d(u,B) < \infty.$$

A and B are *adherent* if they adhere to each other. Let

$$D(A,B) = \max \{\sup_{u \in A} d(u,B), \sup_{v \in B} d(A,v)\}.$$

Then A and B are adherent iff $D(A,B) < \infty$.

Let x be a directed curve, $A \subseteq V\{p,q\}$. x is said to *diverge*$^+$ from A if x gets arbitrarily far away from A moving in the positive direction, i.e. if

$$\sup_m d(x(m,\infty),A) = \infty.$$

Likewise, x *diverges*$^-$ from A if

$$\sup_m d(x(-\infty,m),A) = \infty.$$

x *diverges* from A if it both diverges$^+$ and diverges$^-$ from A. x and y are *divergent* if they diverge from each other.

The notions of divergence$^+$, divergence$^-$, and adherence between curves are not related in general; however,

Proposition 5.2. If x,y are periodic, then the following are equivalent:

i) x diverges$^+$ from y

ii) x diverges$^-$ from y

iii) x does not adhere to y.

In addition,

Proposition 5.3. If x,y are periodic and infinite, then the following are equivalent:

i) x adheres to y

ii) y adheres to x

iii) $Aut(x) \cap Aut(y)$ is nontrivial.

Thus any pair of infinite periodic curves are either adherent or divergent.

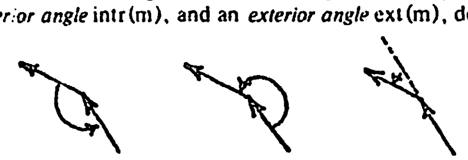
y is said to *approximate* x if $Aut(x) \subseteq Aut(y)$. By the above, if x is infinite and periodic then any curve approximating x adheres to x. Every periodic curve has a simple approximating subcurve.

Curvature

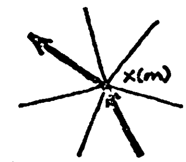
The *total angle* once around each vertex of {p,q} is 2π. The *quantum angle* between two consecutive edges adjacent to a vertex is

$$Q = \frac{2\pi}{q}.$$

Every vertex x(m) along a curve x has a *left interior angle* intl(m), a *right interior angle* intr(m), and an *exterior angle* ext(m), defined by



respectively. For example, the curve



in {3,7} has intl(m) = 3Q, intr(m) = 4Q, and ext(m) = ½Q. The function ext is the discrete analog of *geodesic curvature*. A curve is uniquely determined by two vertices and its exterior angle function.

The *Gaussian curvature* of the graph {p,q} is defined by

$$K = \left(\frac{1}{p} + \frac{1}{q} - \frac{1}{2}\right)\pi$$

The Gaussian curvature of {p,q} has the same sign as the Gaussian curvature of its corresponding surface.

The Gauss-Bonnet Formula

The *Gauss-Bonnet Formula* is one of the central tools of differential geometry. It relates a curve x to the simply connected region S it encloses, in terms of the geodesic curvature g of x and the Gaussian curvature K of the surface. It states that

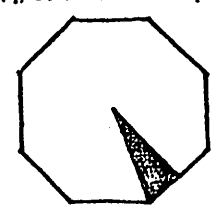
$$\int_x g dx + \iint_S K dS = 2\pi,$$

where the first integral is taken with respect to arc length once around x counterclockwise. For surfaces of constant Gaussian curvature such as spheres and pseudospheres, this reduces to

$$\int_x g dx + KA = 2\pi$$

where A is the area of S.

We have a discrete version of this theorem for the graphs {p,q}. Let a face of {p,q} be divided into 2p congruent triangles, as illustrated.



These triangles are called *fundamental regions* by Coxeter (1963).

Theorem 5.4. Let x be a finite, simple, positively oriented curve in {p,q} with k vertices, enclosing A fundamental regions. Then

$$\sum_{m=0}^{k-1} ext(m) + KA = 2\pi.$$

Proof. Induction on A.

For the remainder of the section, we restrict our attention to the graphs {3,q}, $q \geqslant 6$.

A curve x is *left straight* if

i) for no m does intl(m) = Q;

ii) for no segment x(i,j) does intl(i) = intl(j) = 2Q and intl(m) = 3Q for $i < m < j$.

Right straight is similar. x is *straight* if it is both left and right straight.

If s = x(i,j), define

$$\sum_s ext = \sum_{m=i+1}^{j-1} ext(m).$$

Theorem 5.5. The following are equivalent:

i) x is a shortest path.

ii) x is straight.

iii) for all finite segments s of x,

$$|\sum_{s} ext| \leqslant Q - 6(|s|-1)K.$$

The proofs i) → ii) → iii) are easy. The proof of iii) → i) involves assuming the contrary and deriving a contradiction of 5.4.

Corollary 5.6. Every infinite periodic curve with period n has a straight approximating curve y with $D(x,y) \leqslant n$.

Theorem 5.7. If $q \geqslant 7$ and if x,y are adherent straight curves, then $D(x,y) \leqslant 1$.

The proof is by contradiction of 5.4. Note that 5.7 doesn't hold in {3,6}.

Corollary 5.8. If $q \geqslant 7$ and if x,y are infinite adherent curves with positive periods m,n, then

$$D(x,y) \leqslant 1+m+n.$$

Let $n>0$, $q \geqslant 7$, and let P_n be the set of infinite curves in {3,q} with minimal positive period at most n.

Corollary 5.9. If $x \epsilon P_n$ then all but a finite number of elements of P_n diverge from x.

The following theorem will allow us to pass infinite periodic curves through each other.

Theorem 5.10. Let x,y be periodic straight curves in {3,q}, $q \geqslant 6$. Then there exists $\sigma \epsilon Aut(\{3,q\})$ such that

i) x and $\sigma(y)$ diverge;

ii) $x(0,\infty)$ lies to the left of $\sigma(y)$ and $x(-\infty,0)$ lies to the right of $\sigma(y)$.

The following theorem will help us cut up the graph {p,q} to form a finite graph.

Lemma 5.11. For any translation σ in $Aut(\{3,q\})$, $q \geqslant 6$, there is a curve x such that

i) x and $\sigma \circ x$ are disjoint;

ii) x lies entirely to the left of $\sigma \circ x$ and $\sigma \circ x$ lies entirely to the right of x.

Using 5.11, it is easy to prove

Lemma 5.12. If σ is a translation, then

$$\inf_u d(u,\sigma^k(u)) \geqslant k.$$

Let σ be a translation and let $\{p,q\}/\sigma$ be the graph obtained from {p,q} by identifying vertices u and v iff there is an n such that $v = \sigma^n(u)$. Let $[u]=\{\sigma^n(u) \mid n \epsilon Z\}$. The [u] are the vertices of $\{p,q\}/\sigma$. This graph may be visualized by finding x satisfying 5.11(i) and (ii), cutting {p,q} along x and $\sigma \circ x$, and wrapping the resulting graph around and identifying x(m) and $\sigma(x(m))$ to form a cylinder. Using 5.11 and 5.12, we have

Theorem 5.13. If σ is a translation and $k \epsilon N$, then

i) $\{p,q\}/\sigma^k$ is planar;

ii) the canonical morphism $\{p,q\} \to \{p,q\}/\sigma^k$ which takes $u \to [u]$ is one-one on regions of {p,q} of diameter k;

iii) the map $[u] \to [\sigma(u)]$, denoted $[\sigma]$, is an automorphism of $\{p,q\}/\sigma^k$.

5.13(ii) says that $\{p,q\}/\sigma^k$ looks locally like {p,q} on regions of diameter $\leqslant k$.

If H is a subgraph of {p,q}, let [H] be its image under the canonical morphism $\{p,q\} \to \{p,q\}/\sigma^k$. The following is immediate from 5.13(iii).

Corollary 5.14. If H is a subgraph of {p,q} preserved by σ, then [H] is preserved by $[\sigma]$.

6. A lower bound for three automata in planar graphs

In this section we combine the results of the previous two sections and prove

Theorem 6.1. No three finite automata together can search all finite cubic embedded graphs.

We will be discussing the motion of 3 automata on $\{7,3\} \circ \phi$ for some suitable hypernode ϕ. In order to use the theory of section 5, we will view periodic curves in $\{7,3\} \circ \phi$ as periodic curves in {3,7}, by fixing once and for all a pair of morphisms

$$\{7,3\} \circ \phi \xrightarrow{h} \{7,3\} \xrightarrow{g} \{3,7\}$$

where h collapses ϕ and g associates vertices (faces) of {7,3} with faces (vertices) of {3,7}. $\{7,3\} \circ \phi$, {7,3} and {3,7} are all in a sense isomorphic, in that h and g provide one-one correspondences between hypernodes (hyperedges, hyperfaces) of $\{7,3\} \circ \phi$, nodes (edges, faces) of {7,3}, and faces (edges, nodes) of {3,7}; moreover, h and g induce isomorphisms

$$Aut_\phi(\{7,3\} \circ \phi) \xrightarrow{h} Aut(\{7,3\}) \xrightarrow{g} Aut(\{3,7\}).$$

Thus, periodic curves x in $\{7,3\} \circ \phi$ may be viewed as periodic curves g(h(x)) in {3,7} which mimic x. We will use this correspondence freely in the proof.

Proof of Theorem 6.1. Let A, B, C be 3 automata. Let $\phi = \phi_A \circ \phi_B \circ \phi_C$, where ϕ_A is defined in 4.5. By 4.2, 4.4, and 4.5, A, B, and C are each individually degenerate in $\{7,3\} \circ \phi$.

Consider the motion of A and B together in $\{7,3\} \circ \phi$, in the absence of C. At any point in time, either

(i) A and B will land on the same edge at some time in the future; or

(ii) A and B will never see each other again.

If motion (i) occurs indefinitely, then the motion of A and B together is eventually periodic, since they may occupy the same edge in at most $|A||B|$ pairs of states, and since any subset of $2(|\phi|+4)$ directed edges has 2 similar edges under $Aut_\phi(\{7,3\} \circ \phi)$ (a pair of edges are *similar* if there is a σ mapping one to the other). Thus the motion of A and B together is modeled by a periodic curve in {3,7}.

If motion (ii) occurs eventually, then the motion of A and B is degenerate; they each trace circles in $\{7,3\} \circ \phi$, making no progress.

Now consider the motion of A, B, C together in $\{7,3\} \circ \phi$. At any point in time, either

(iii) each of A, B, C will see one or both of the other two automata at some point in the future;

(iv) C will never see A or B again, but A and B will see each other again (or some other permutation of A, B, C);

(v) no pair of A, B, C will see each other again.

If (iii) occurs indefinitely, then at any time some 2 automata are within 2 hyperfaces of each other, and infinitely often, all 3 are no more than 2 hyperfaces apart. This says that the motion of A, B, C together is eventually periodic. If (iv) eventually occurs and then occurs indefinitely, then C is degenerate, while A, B are governed by (i) or (ii) above. Otherwise, if (v) occurs eventually, then all 3 automata are degenerate.

Now we will construct a finite embedded graph such that A, B, C cycle in the graph without visiting every vertex.

Start the automata together in $\{7,3\} \circ \phi$. If (v) eventually occurs, then only a finite portion of the graph is ever visited, and the argument of 4.1 applies. If (iii) occurs indefinitely, then the three automata together never get very far apart, and trace a periodic path; thus the argument of 4.1 applies (use 5.13 and 5.14 to form a cylinder such as that appearing in 4.1). Otherwise, (iv) occurs eventually, and then occurs indefinitely; i.e., A, B, C run together for a while, then A, B move away from C. A and B eventually trace a periodic path, and can never get more than one hyperface apart, otherwise (ii) and hence (v) occurs. Then the path of A and B together is modeled by a periodic curve x in {3,7}. If x is finite, then only a finite portion of the graph is ever visited, and the argument of 4.1 applies. Otherwise x is infinite, and A and B get arbitrarily far away from C (Theorem 5.1(iii)), since C is degenerate.

Let x be parametrized so that x(0) is near C and so that A and B are moving in a positive direction along x. Let k be a period of x. When A and B get sufficiently far from C, say at x(nk), pick them up, move them to x(−nk), and put them down in the same configuration. They will move in the positive direction along x until they reach x(0) again. If they don't see C again, so much the better. If they see C again, motion (iii) is resumed.

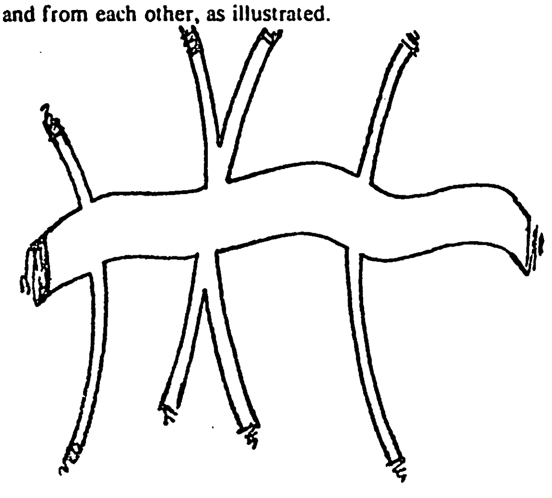
Later on, we will connect x(nk) and x(−nk) for some very large n, in order to form a large periodic loop which looks locally like x. Right now the purpose of moving A and B by hand from x(nk) to x(−nk) is to see what the automata will do next, when the three of them meet again near x(0).

If (iv) happens again, say this time B and C move away from A on an infinite path, we repeat the above process -- extend the path of B and C backward, and move them to the back end of this path, so that they will be moving toward A.

Assume that, after being moved by hand, the two automata we have moved always meet up again with the third automaton, and the subsequent motion of the three automata together eventually results in motion (iv). This is the hardest case; the other cases (motion (iii) indefinitely; motion (v)) will be left to the reader.

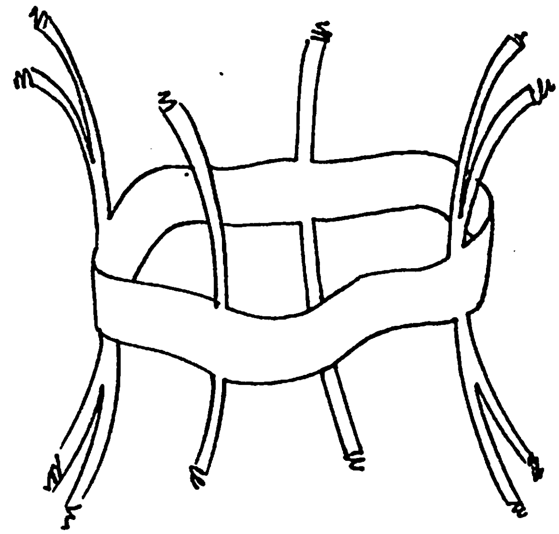
Under the assumptions we have made, we must move two automata by hand infinitely often. Eventually, the motion of A, B, C, together with our removal of pairs of automata by hand, becomes periodic. We define the *main band* as the set of vertices visited by automata during motion (iii). The main band is the path taken by the 3 automata when they are all together. At times two automata leave the main band on an infinite path, leaving the third automaton on the main band. These two automata are then said to be on a *side band*. The main and side bands are all modeled by periodic curves in {3,7}. The main band may be finite (i.e. may trace a closed circuit); again, this is the easier case, so we will leave this case to the reader and assume that the main band is infinite.

If one half of a side band diverges from the main band, the other half does as well (Proposition 5.2). If a side band adheres to the main band, we can just include it in the main band. There are only finitely many of these (Corollary 5.9). Also, only a finite number of paths of pairs of automata can occupy the same side band, and distinct side bands diverge (Corollary 5.9). Thus we have a periodic main band with periodic side bands diverging from it and from each other, as illustrated.

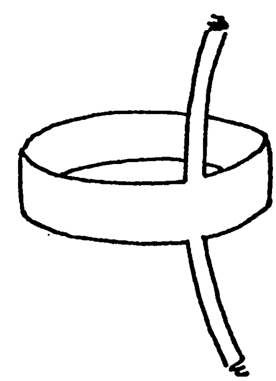


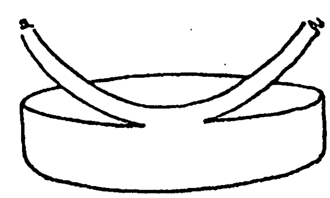
We now wish to separate the graph and connect the ends of the main and side bands so that each band forms a continuous periodic cycle. Then A, B, C will go around this graph forever without visiting every vertex: whenever the three automata are together, they are on the main band; when 2 automata split away from the third, they trace a side band around and back to the main band, and the three automata resume tracing the main band.

First we cut the graph and connect the ends of the main band, making sure it is several periods long. This is done by forming $\{3,7\}/\sigma^k$ for some large k, where $\sigma \in$ Aut(main band) (Theorem 5.13). Since the subgraph of {3,7} consisting of the main and side bands is preserved by σ, Corollary 5.14 gives us the following situation:

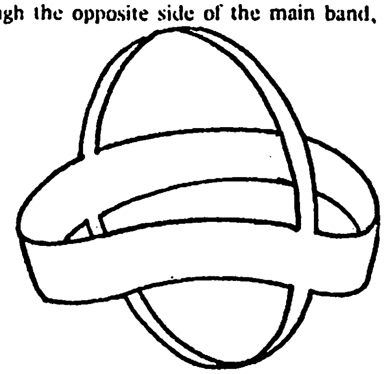


It would be easy to connect the ends of the side bands to form a toroidal graph, by just bringing them around the outside and attaching them. However, we must make the graph planar. Since all side bands are periodic, we can make them as long as we like (there are never any synchronization problems). There are two cases:





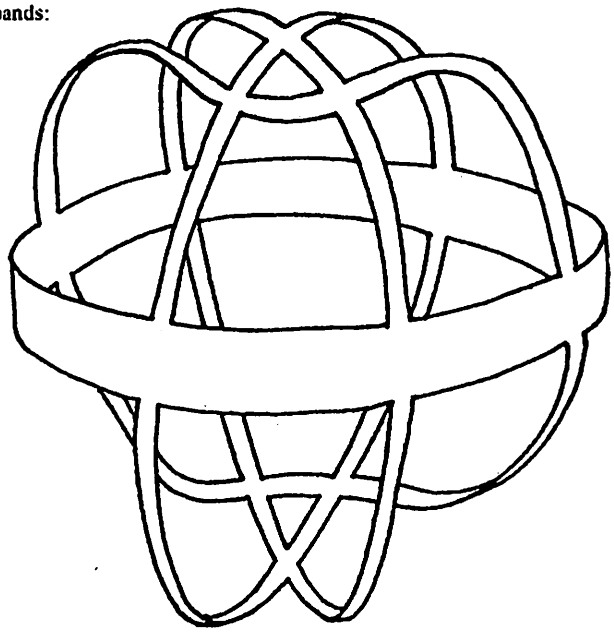
In the first case, we will bring the top of the side band around, down through the opposite side of the main band, and then attach the ends:



This will entail passing side bands through each other and through the main band. Corollary 5.6 and Theorem 5.10 allow us to do this. Passing the side band through the opposite side of the main band may be viewed as another side band emanating from the main band at that point. Such extra side bands exist by Theorem 5.10.

In the second case, the construction is the same, except there is no need to pass through the main band.

If we do this for every side band, the result can be embedded on a sphere. Below is a picture of a main band with three side bands:



This completes the proof.

Conclusion

The behavior of finite automata on strings is thoroughly understood. In light of this, we find it quite intriguing that lower bounds for their behavior in graphs and mazes should be so difficult. The authors, together with David Lichtenstein, worked for some time on a lower bound for a single automaton in mazes before discovering Budach's proof. Although he was several ideas ahead of us, we were going in the same direction, after exhausting many other tracks. This indicates to us that Budach's proof is the right one, and unlikely to be simplified.

We are also quite excited by the wealth of interesting relationships which we have observed in conducting this research, many of which in our opinion merit investigation. In particular we would like to mention the following open problems:

(i) Is it possible to construct a 1-counter automaton that, when placed on a white cell next to a barrier will walk once completely around the boundary of that barrier, and halt back where it started?

(ii) Our 1-counter algorithm operates in time $O(n^2)$ and space $O(\log(n))$. A linear (space-) bounded automaton can search an arbitrary maze in linear time, by constructing a map. Is there any algorithm for searching mazes in linear time using only $O(\log(n))$ space, or even $o(n)$ space? Such a machine would not have enough space to construct a map.

(iii) Coy (1977) has shown that no pushdown automaton can search all finite planar cubic graphs, by showing that in the absence of information from the graph, the pushdown store gives no extra power. Is there a formal relationship between this result and the analogous fact that all context free languages over a single letter alphabet are regular?

(iv) Give a finite set of finite automata which together can search all finite cubic planar graphs, or show no such set exists (Paul and Tarjan have conjectured the latter). The lower bound proof for two and three automata involves constructing a graph in which two automata act like one automaton and one automaton acts like a pebble. So far we have been unsuccessful in generalizing this technique.

(v) Give a 1-pebble automaton that can search all mazes, or show that none exists.

Acknowledgment

We are deeply indebted to David Lichtenstein and Bruce Char for numerous ideas and helpful discussions.

Bibliography

Beyer, T., "Recognition of topological invariants by iterative arrays," Project MAC TR-66, MIT, 1966.

Blum, M. and C. Hewitt, "Automata on a two-dimensional tape," Proc. 8th IEEE SWAT Conf. (1967), 155-160.

Blum, M. and W. Sakoda, "On the capability of finite automata in 2 and 3 dimensional space," Proc. 17th IEEE FOCS Conf. (1977), 147-161.

Brahana, H.R., "Certain perfect groups generated by two operators of orders two and three," Amer. J. Math. 50(1928), 345-356.

Budach, L., "Automata and labyrinths," unpublished manuscript, 1974.

Budach, L., "On the solution of the labyrinth problem for finite automata," EIK 11(1975), 10-12, 661-672.

Budach, L., "Environments, labyrinths, and automata," in: *Fundamentals of Computation Theory*, ed. Karpiński (Berlin: Springer-Verlag, 1977), 54-64.

Coxeter, H.M.S. *Regular Polytopes.* New York: Macmillan, 1963.

Coxeter, H.M.S. and W.O.J. Moser. *Generators and Relations for Discrete Groups.* Berlin: Springer-Verlag, 1972.

Coy, W., "Automata in labyrinths," in: *Fundamentals of Computation Theory*, ed. Karpiński (Berlin: Springer-Verlag, 1977), 65-71.

Döpp, K., "Automaten in labyrinthen," EIK 7:2(1971), 79-94; 7:3(1971), 167-190.

Miller, G.A., "Groups defined by the orders of two generators and the order of their product," Amer. J. Math 24(1902), 96-100.

Müller, H., "Endliche automaten und labyrinthen," EIK 7:2(1971), 261-264.

Mylopoulos, J., "On the recognition of topological invariants by 4-way finite automata," Computergraphics and Image Processing 1(1972), 308-316.

Rosenfeld, A., "Connectivity in digital pictures," J. ACM 18(1970), 146-160.

Shah, A.N., "Pebble automata on arrays," Computergraphics and Image Processing 3(1974), 236-246.