The Rotation Group of Rubik's Cube

Oliver Knill

Roman E. Mäder

ETH Zürich

Revised by Robin S. Sanders and Julie D. Simon

University of Illinois at Urbana-Champaign

ABSTRACT

In this project, the group theory program "Cayley" is used to solve some problems associated with Rubik's Cube, including the original problem of restoring the initial state of the cube.

Project-No.:	003-021
Title:	Rubik´s Cube
Subject:	Algebra
Programs:	Cayley
References:	001-021 Die Drehgruppe des Rubikwürfels
Typesetting:	mltroff
Version:	5/15/87

1. Introduction

Several years have passed since the Rubik's Cube conquered the world. Almost everyone knows this cube and many have solved it or have tried to do so for hours. Rarely has there been a puzzle which has caused such a stir and which has found its way not only into the world of games but also into that of mathematics and computers.

Before you start your work on this project about Rubik's cube, we recommend that you reach for a cube and get familiar with it (again). Many concepts of group theory can be demonstrated with this cube, and abstract definitions can become more intuitive.

1.1. The Cube

If you already know it or are even a cube expert, you may skim this section which defines the structure of the cube and also introduces some notation.

The Rubik's cube consists of $3\times3\times3 = 27$ smaller cubes arranged in one larger cube. The 54 visible faces are colored such that in the initial state, the 6 faces of the large cube are each a different color. Each layer of $3\times3 = 9$ cubes can be rotated by multiples of 90^0. After a few rotations, the colors are in disorder and it is nearly impossible to rearrange them by accidental rotations. The exact number of states that the cube can assume will be one of the results of *this project*.

1.2. Notation

Lower case letters denote the 6 faces of the large cube, as follows:

f (front), b (back), u (up), *d* (down), *l* (left), *r* (right)

Clockwise 90^0 rotation of a lateral 3×3 layer is denoted by the corresponding upper case letters *F, B, U, D, L and R.* Rotations by 180^0 are denoted by F^2, etc. and counter-clockwise 90^0 rotations by F^{-1}, etc.

As an example, the product $LDL^{-1}U^{-1}DLD^{-1}LUD^{-1}$ is a cyclic permutation of three "edge cubes".

2. The Group of the Cube as a Permutation Group

For mathematically dealing with the cube. we also need a formal description of it. The adequate tool is *permutation groups.*

We start by enumerating the visible faces of the smaller cubes, ignoring the ones in the center of each face of the larger cube. (The latter can be considered as fixed points if we agree never to rotate a central 3×3 layer, which is not a real constraint.)

The rotations can now be written as permutations, where each permutation is represented by a product of cyclic permutations:

U = (1, 3, 8, 6)(2, 5, 7, 4)(9, 48, 15, 12)(10, 47, 16, 13)(11, 46, 17, 14)

L = (9, 11, 26, 24)(10, 19, 25, 18)(1, 12, 33, 41)(4, 20, 36, 44)(6, 27, 38, 46)

F = (12, 14, 29, 27)(13, 21, 28, 20)(6, 15, 35, 26)(7, 22, 34, 19)(8, 30, 33, 11)

R = (15, 17, 32, 30)(16, 23, 31, 22)(3, 43, 35, 14)(5, 45, 37, 21)(8, 48, 40, 29)

D = (33, 35, 40, 38)(34, 37, 39, 36)(24, 27, 30, 43)(25, 28, 31, 42)(26, 29, 32, 41)

B = (41, 43, 48, 46)(42, 45, 47, 44)(1, 24, 40, 17)(2, 18, 39, 23)(9, 38, 32, 3)

The rotation group $G3$ of the cube is now the group *generated* by U, L, F, R, D and B. We write $G3 = \langle U,L,F,R,D,B \rangle$. The *degree* of $G3$ is 48.

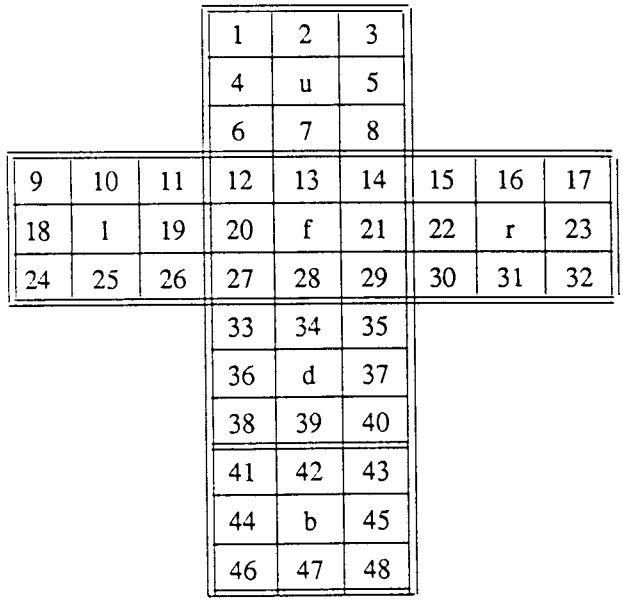

Fig. 2-1

3. Problems

3.1. "Subgroups"

First, we want to familiarize ourselves with the Cayley program:

Find the order of the following subgroups:

a) The Rubik group $G3$ itself

b) The "square group" $\langle R^2, L^2, F^2, B^2, U^2, D^2 \rangle$

c) The "bridge group" $\langle R, U, L \rangle$

d) The "corner group" $\langle R, U, F \rangle$

e) The "antislice group" $\langle LR, UD, FB \rangle$

f) The "slice group" $\langle RL^{-1}, UD^{-1}, FB^{-1} \rangle$

A permutation is called *central* if it commutes with every other permutation. Verify by theoretic reasoning that the set of all central elements of a group is a subgroup, and even a normal divisor, called the *center* of the group. Use Cayley to find the center of:

a) the Rubik group $G3$

b) The "edge group" $\langle U, R \rangle$

c) The "slice group" $\langle RL^{-1}, UD^{-1}, FB^{-1} \rangle$

and find an interpretation of the result on your Rubik's cube.

You can easily verify with Cayley that each 5 of the 6 generators suffice to generate $G3$. Find, by experimenting, a system of as few generators as possible. (There exists one with only two generators).

Optional:

The *symmetric group of order n* consists of all permutations of n elements. Find by theoretic reasoning all symmetric subgroups of $G3$.

Optional:

A *cyclic group of order n* is a group having a single generating element c, such that c^n is the identity. Try to get an overview over all cyclic subgroups of $G3$.

3.2. "Structure"

Now, we want to investigate the structure of $G3$. Cayley can help us to accomplish this.

Find the *orbits* of $G3$ and find also the groups that result if the group operations are restricted to these orbits. Find an interpretation for the orbits on your cube.

For each one of these groups find now the *minimal block system*, i.e. those packets of points that always stay together, and interpret them on your cube.

Find out what groups deal only with the packets.

Optional:

Prove (by theoretical reasoning) that the following permutations are not in $G3$, i.e. that the described situations can not be obtained from the initial state by rotations:

a) Only one single "corner cube" or "edge cube" is twisted

b) Only two "corner cubes" or "edge cubes" have changed places

3.3. "Strong Generating Sets"

When they first encountered the Rubik's cube, many people spent large amounts oftime finding a method to solve it. Then each time they discovered a similar puzzle, they found the new one a little easier to solve than the last. Some of these other puzzles are: Rubik versions of the other regular polyhedra, the 4×4×4 cube, the magic drum and the tower of Babylon.

The fact that experience makes the solution easier indicates that there might be an algorithm for systematically solving all of these problems. Such an algorithm indeed exists and it is based on the so-called *strong generating sets.*

Let G be some permutation group operating on the set $M = \{1, \ldots, n\}$. The *stabilizer of the point* $i \in M$ is the set of those elements in G that fix i . The *pointwise stabilizer of the subset* $A \subseteq M$ is the set of those elements in G that fix each element of A . Formally this is:

$$Stab(i) = \{g \in G: g(i) = i\} \tag{3-1}$$

$$Stab(A) = \{g \in G: g(a) = a \forall a \in A\} \tag{3-2}$$

Note the pointwise stabilizer for A is not the usual definition for stabilizer of a subset, which is:

$$Stab(A) = \{g \in G: g(a) \in A \forall a \in A\} \tag{3-3}$$

All of the stabilizers that we will use are pointwise stabilizers, and the notation, $Stab(A)$, will always refer to the pointwise stabilizer of A.

Obviously, $Stab(A)$ is a subgroup of G. The *orbit* of the point $i \in M$ is the set of those $m \in M$, such that there is an $g \in G$ which maps i to m. That is:

$$Orb(i) = \{m \in M: \exists g \in G\ g(i) = m\} \tag{3-4}$$

$$Orb(A) = \{m \in M: \exists g \in G\ \exists a \in A\ g(a) = m\} \tag{3-5}$$

What are the stabilizers in our Rubik group G? Most strategies to solve Rubik's cube reveal that there is a sequence of steps such that each step must fix more of the small cubes than the previous step did. For example, most people first fix one 3×3 layer and then try to keep this layer fixed. Of course, a step is usually not an application of one of the generators L, R etc., but a product of such. The last step in the sequence is typically one which

performs a cyclic permutation of three "edge cubes" and which fixes all the other cubes.

These strategies use the fact that the stabilizers of the sets {1,2,...,k} of integers are nested as k increases. Using the symbol ">" for the subgroup relation, we can illustrate this fact:

$$G = Stab(\{\}) > Stab(\{1\}) > Stab(\{1, 2\}) > \ldots Stab(M) = E \tag{3-6}$$

where E is the trivial group. Next, we must find all of those permutations which are allowed *in a particular stabilizer*. For that, we need a set of generators of $G3$ whose intersection with $Stab(\{1, 2, \ldots, k\})$ generates this stabilizer (for any $1 \le k \le n$). Such a set is called *strong generating set (SGS)*.

Why does a SGS constitute a solution for Rubik's cube? Let g be some given state of the cube, i.e. some permutation of the set $M = \{1, 2, \ldots, 48\}$. Then, g moves the point 1 to some point $i \in M$. We now find a word s over the SGS which moves 1 to i. The product gs^{-1} is then an element of $Stab(1)$. That means, after applying the permutation s^{-1} to the current state of the cube, 1 will be in the correct place.

We now intersect the SGS with $Stab(1)$. This gives us a SGS for the subgroup $Stab(1)$, to which we will now restrict ourselves. We find a word h over this new SGS that brings 2 to its current place. Multiplying the current permutation gs^{-1} with the inverse of h produces the element $gs^{-1}h^{-1}$ of $Stab(\{1, 2\})$

We iterate this procedure until we reach $Stab(M)$. Then, the cube is completely restored.

Computing a Strong Generating Set

Strong generating sets can be computed with an algorithm due to Charles Sims. The method is the following:

Let S be a given set of generators. For each $b \in Orb(1)$ compute some $r_b \in G$ which maps 1 to b. For each $s \in S$ and for each $b \in Orb(1)$ the so-called *Schreier generators*

$$S(s, b) = r_b s r_{s(b)}^{-1} \tag{3-7}$$

are computed. By the following theorem, these Schreier generators generate $Stab(1)$. Choosing the Schreier generators as the new set S, the procedure can now be repeated for the element 2 and we get a set of generators for $Stab(\{1, 2\})$, etc.

Schreier's Theorem

> Let G be a permutation group on $M = \{1, \ldots, n\}$ with a set $S = \{s_1, \ldots, s_m\}$ of generators and let $a \in M$. For every $b \in Orb(a)$ let r_b denote an element of G with $r_b(a) = b$. Finally, define $S(s, b)$ as in equation 3-7.
>
> Then $\langle S(s, b): s \in S, b \in Orb(a) \rangle = Stab(a)$

The proof is not given here, although it is not difficult. The SGS for G is the union of the sets of Schreier Generators for the groups shown in (3-6). We can now work on solving the cube.

The problem of restoring Rubik's Cube

Let G be a permutation group generated by the set $S = \{g_1, \ldots, g_k\}$. Note that S is not a SGS for G. Write a Cayley procedure which will generate a SGS for G whose elements are words over the set S. (A *word over the set S* is simply a product of elements in S that Cayley has not evaluated, i.e. it is written with names of elements not permutations.) Then write a Cayley procedure which, given a permutation $p \in G$, returns a word w over the set S that tells you how to undo the permutation p, i.e. when the original permutation p is multiplied succesively by the generators in the word w, the identity permutation results.

You should run your procedures on the subgroups *slice* and *antislice* of the Rubik's Cube group, $G3$. For each of these subgroups you will need to create a library file containing the group along with its generators.

Some hints: You may want to use the symmetric group of order 5 to develop your procedures. The symmetric group of order 5 is generated by the permutations *five* = (1, 2, 3, 4, 5) and *two* = (1, 2).

In the library for this project you will find three Cayley procedures, *plan*, *sims*, and *how*, to help you with this problem. You may use these procedures in your solution.

You should use *plan* and *sims* to write a Cayley procedure that will generate a set (or a sequence) of Strong Generators for a permutation group G generated by the set S. The elements in your Strong Generating Set should be words over S. You can then write a Cayley procedure that will undo permutations in G.

When you are writing your procedure to undo permutations, you may want to use a modified version of *plan* called *how*.

The three procedures:

The procedure plan

Given a permutation group G, a set of words, S, that generate a subgroup $H = \langle S \rangle$ of G, and an integer a in the set on which G acts, this procedure returns a sequence *bus* of words in G. If $b \in orbit(\langle S \rangle, a)$ then bus[b] is a word that maps a to b. The procedure is shown below.

```
library plan;
procedure plan(G,S,a;bus);
&       G is a group that contains a subgroup that is generated by the
        words is S.
        S is a set of words that generate a subgroup, H=<S>, of G.
        a is an element of the set on which H operates.
        Then in bus a sequence of elements of H is returned.
        These elements are given as products of elements of S
        (i.e. as words over S).
        They indicate how the elements in the orbit of a can be reached
        from a.
&
   bus=empty;
   for n=1 to degree(G) do
     bus[n]:=identity of G;
   end;
   orb=orbit(<S>,a);
   orb=orb-[a];
   if orb eq [ ] then return; end;
   ta:=[ ];
   tried=[ ];
   for each x in S do
      x1=evaluate(G,x);
      if (a^x1 ne a) then
        ta:=ta join [x];
        tried=tried join [x1];
      end;
   end;
   bahn=empty;
   while orb ne [] do
       for each x in ta do
            x1=evaluate(G,x);
            c=a^x1;
            if c in orb then
              bahn=append(bahn,x);
              orb=orb-[c];
```

```
            if orb eq [] then break;
           end;
          end;
      end;
      if orb eq [] then break;
      end;
      temp:=[ ];
      for each y in ta do
          for each x in S do
          x1=evaluate(G,y*x);
          if not(x1 in tried) then
            temp:=temp join [y*x];
            tried=tried join [x1];
          end;
          end;
      end;
      ta:=temp;
   end;
   for n=1 to length(bahn) do
       bus[a^evaluate(G,bahn[n])]=bahn[n];
   end;
end;
finish;
```

The procedure sims

This procedure uses Sims's algorithm to find the Schreier generators for a particular stabilizer.

Given a permutation group G, a set of words, *gen*, that generate a subgroup $H = \langle gen \rangle$ of G, an element z in the set on which G acts, and the sequence R, the output of plan(G,*gen*,z;R), this procedure finds the Schreier generators for *Stab*(H,z), the stabilizer of z in the subgroup H. It returns the *words* that generate *Stab*(H,z) in the sequence $S1$. It returns the *permutations* that generate *Stab*(H,z) in the set $T1$.

The procedure *sims* is shown below.

```
library sims;
procedure sims(G,gen,R,z;S1,T1);
&     Input to sims
  G is a permutation group.
  gen is a set of words that generate a subgroup of G.
  R is the output of procedure plan.  For each j in the orbit of z,
    R contains one map, R[i], that sends z to j.
  z is in the set that G acts on.
      Output from sims
  S1 is a sequence of words in gen that generate Stab(<gen>,z).
  T1 is the set of permutations in S1.
&
S:=setseq(gen);
e:=identity of G;
S1:=seq(e) of G;
T1=[identity of G];
G1=<T1>;
orb=orbit(<gen>,z) - [z];
if orb eq [ ] then
  S1:=S;
```

```
  T1=evaluate(G,gen);
  return;
end;
ent=1;
for j=1 to length(S) do
   sj=evaluate(G,S[j]);
    for each i in orb do
       ri=evaluate(G,R[i]);
       d=z^ri;
       b=d^sj;
        for k=z to length(R) do
          f=evaluate(G,R[k]);
          p=z^f;
          if (p EQ b) then
              RB1:=R[k];
              rb2=evaluate(G,R[k]);
              break;
           end;
         end;
        x:=R[i]*S[j]*RB1^(-1);
        y=evaluate(G,x);
        c=z^y;
         if not(y in G1) then
           T1=T1 join [y];
           S1[ent]:=x;
           G1=<T1>;
            ent=ent+1;
         end;
    end;
   end;
end;
finish;
```

The procedure how

This procedure is a modified version of *plan*. Unlike *plan*, *how* requires two input integers, *a* and *b*. The output, *bahn*, is a single word in *G* with the property that *bahn* takes *a* to *b*. You may find this procedure useful if you run into memory problems (or time problems) when you are writing your procedure to undo permutations in *G*.

The procedure is shown below.

```
Library how;
procedure how(G,S,a,b;bahn);
&       G is a group that contains H=<S> as a subgroup.
        S is a set of words in G that generate a subgroup H.
        a is an element of the set on which H=<S> operates.
        b is an element of the orbit of a in the subgroup H.
        bahn indicates how the element b can be reached from a.
        Note that bahn is given as a product of elements of S,
        (i.e. bahn is a word over S).
&
```

```
if not (b in orbit(<S>,a)) then
     print b,'is not in the orbit of',a;
     return;
end;
if (b eq a) then
     bahn:= identity of G;
     return;
end;
tb:=[ ];
tried=[ ];
for each x in S do
x1=evaluate(G,x);
  if (b^x1 ne b) then
  tb:=tb join [x];
  tried=tried join [x1];
  end;
 end;
 flag=true;
 while (flag eq true) do
   for each x in tb do
     x1=evaluate(G,x);
     c=a^x1;
     if (c eq b) then
        bahn:=x;
        flag=false;
        return;
     end;
    end;
    temp:=[ ];
    for each y in tb do
       for each x in S do
       x1=evaluate(G,x*y);
         if not(x1 in tried) then
         temp:=temp join [x*y];
         tried=tried join [x1];
         end;
       end;
    end;
    tb:=temp;
  end;
end;
finish;
```

4. Optional Problems

4.1. "The Geometry of Rubik's Cube"

The Rubik group $G3$ can be regarded as a geometric object. For that, we introduce the notions:

i) The *distance* between two elements of $G3$ is the smallest number of rotations of lateral (that is, not central) 3×3 layers that leads you from the first to the second element. Examples of such rotations are D, D^2 or D^{-1}.

ii) A *circle* is a set of elements that are a fixed distance r from some fixed element P (the center of the circle). A *disk* is obtained if the distance from P is less than or equal to r. The elements of this disk(or circle) set are permutations, elements of the group. Think of starting with the cube in its solved state and then making r twists of the sides. The permutation you get will be an element of the circle of radius r with center at the identity.

iii) The *length* of the group $G3$ is the radius of the smallest disk centered at the identity that covers $G3$.

Of course, further geometric notions could now be introduced. However, with only these simple ones, we can ask some difficult questions about Rubik's Cube. Presently,it is known that the *length* of $G3$ is at most 23, and conjecture that it is 17.

a) Find the *circle* with center at the identity and radius 5 in the "edge group" $<U, R>$ and one that returns the *disk* with the same features. It is simpler, for the purposes of this problem, to consider only single, clockwise rotations. That is, use only the generators themselves.

b) Prove that the length of $G3$ is at least 16. (In this problem, you should consider also squares and inverses of generators as rotations.)

4.2. "The Rubik Tetrahedron"

You will find the group *tetra* as a file in the library of this project. This group is of *degree* 36 and is simpler than the cube. If the enumeration is chosen as in fig. 4-1, the set of generators may be chosen as:

A = (26,3,17)(8,15,24)(16,25,4)(27,5,18)

B = (3,35,12)(6,30,15)(2,31,11)(1,32,10)

C = (12,33,24)(30,21,17)(20,13,29)(14,28,19)

D = (26,33,6)(8,21,35)(22,34,7)(23,36,9)

You may notice that the corners do not appear in any of the generators. The corner pieces can only be rotated on a central pin. They cannot be moved away from their respective neighbors. No twist of a tetrahedron slice will change the position of a corner piece. Therefore, they may be thought of as fixed pieces, similar to the center squares on the Rubik's cube.

You will not be able to run the procedures you wrote for the "Strong Generating Sets" problem on *tetra*. The output generated for this group by the procedures is too large for the space available in cayley.

a) Find the order of *tetra*.

b) Find the center of *tetra*.

c) Find the orbits of *tetra*, the groups that result if the group operations are restricted to these orbits, and the block system for each of these groups. (Use the methods of problem 2.)

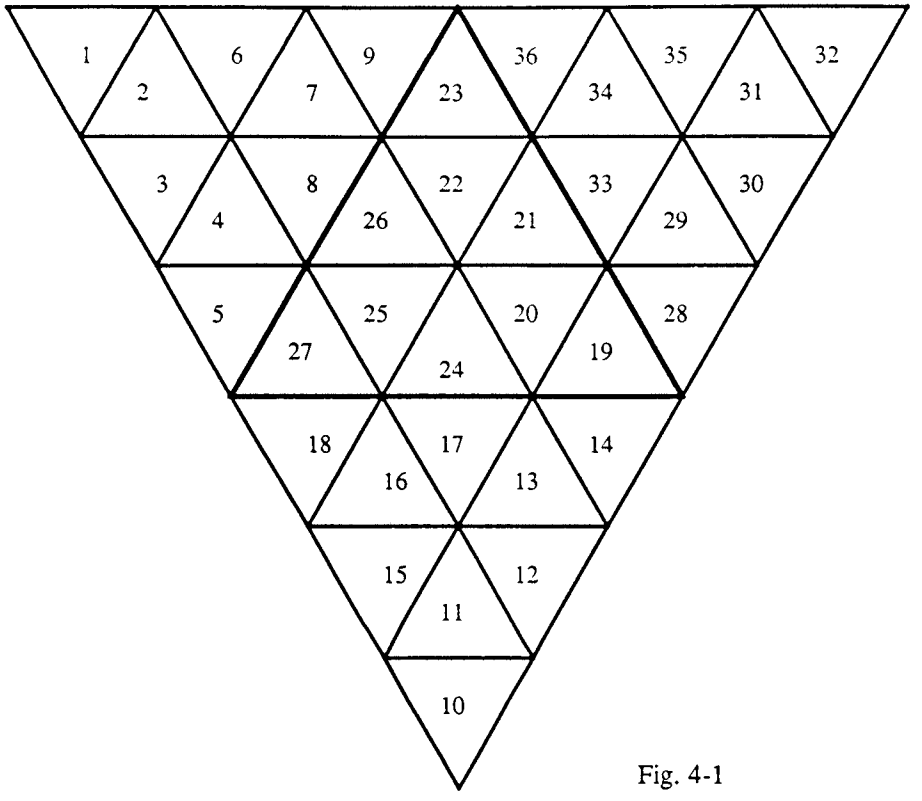

Fig. 4-1

References

1. C. Bandelow, *Einfuehrung in die Cubologie*, Vieweg, 1981.

2. C. Bandelow, *Inside Rubik's Cube and Beyond* , Birkhauser, Boston, 1982. English translation of the above book.

3. John J. Cannon, *A language for group theory: CAYLEY-Reference Manual*, University of Sydney, Dec. 1982.

4. J. Ewing and C. Kosniowski, *Puzzle It Out: Cubes, Groups, and Puzzles*, Cambridge University Press, 1982.

5. A. H. Frey, Jr. and D. Singmaster, *Handbook of Cubik Math*, Enslow Publishers, Hillside, New Jersey, 1982.

6. D. R. Hofstadter, "Metamagical Themas - The magic cube's cubies are twiddled by cubists and solved by cubemeisters," *Scientific American*, March 1981.

7. D. R. Hofstadter, "Vom Zauber des Zauberwuerfels ," *Spektrum der Wissenschaft (German edition of the Scientific American)*, May 1981. German translation of the above article.

8. D. R. Hofstadter, "Metamagical Themas - Beyond Rubik's Cube: spheres, pyramids, dodecahedrons and God knows what else," *Scientific American*, July 1982.

9. D. R. Hofstadter, "Die Ausgeburt des Wuerfels: eine teuflische Sippschaft ," *Spektrum der Wissenschaft (German edition of the Scientific American)*, September 1982. German translation of the above article.

10. D. Singmaster, *Notes on Rubik's Magic Cube*, Enslow Publishers, Hillside, New Jersey, 1981.

ACM PRESS

ASSOCIATION FOR COMPUTING MACHINERY · FOUNDED 1947 ·

SIGSAM Bulletin

A Quarterly Publication of the
Special Interest Group on Symbolic & Algebraic Manipulation

Volume 21, Number 3 August 1987 (Issue #81)