Performance of the GAP-function Normalizer and an attempt of its improvement

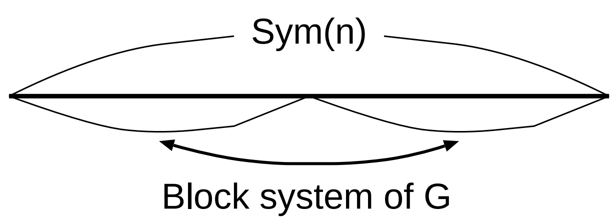
Izumi Miyamoto
University of Yamanashi
imiyamoto@yamanashi.ac.jp

1 Introduction

Let $\Omega = \{1, 2, \cdots, n\}$. Let G and H be permutation groups on Ω. The normalizer of G in H is defined by

$$\text{Normalizer}(H, G) = \{\ h \in H | h^{-1}Gh = G\ \}.$$

Suppose $H = Sym(n)$ =SymmetricGroup(Ω). GAP4r4[1] - Groups, Algorithms, Programming (version 4)- a System for Computational Discrete Algebra has a special function in this case. If G is intransitive or imprimitive, the normalizer is computed in a smaller subgroup of $Sym(n)$.



If G is as above, let W =WreathProduct($Sym(n/2), Sym(2)$), then

$$\text{Normalizer}(Sym(n), G) \subseteq W$$

So GAP4 computes Normalizer(W, G) in this case. Even if G is primitive, we have such a subgroup K like above in [2], if G is not doubly transitive.

Proposition 1.1 *[2] If G is transitive, then the normalizer of G is contained in the automorphism group of the association scheme formed by G, which consists of the orbits of G on $\Omega \times \Omega$.*

If G is transitive, we can also use the following lemma.

Lemma 1.2 *[3] Let K be a permutation group on Ω. Let F be a tuple $[p_1, p_2, \cdots, p_r]$ of points in Ω and let G^i be the stabilizer of the subset $[p_1, p_2, \cdots, p_i]$ of F as a* *tuple in G for $i = 1, 2, \cdots, r$. Let I^i be the group of isomorphisms of the system of association schemes of G^i on $\Omega \backslash [p_1, p_2, \cdots, p_i]$. Set $I^0 = I$, $G^0 = G$ and set $I^{\{0..i\}} = I^0 \cap I^1 \cap \cdots \cap I^i$. Suppose that $G^i \cap K$ is transitive on the orbit of $I^{\{0..i\}} \cap K$ containing the point p_{i+1} for $i = 0, 1, \cdots, r-1$. Then the normalizer of G in K is generated by $G \cap K$ and the normalizer of G in $I^{\{0..r\}} \cap K$.*

In [3, 4] the author wrote a program to compute the normalizers of permutation groups using these algorithms. Especially, the program computes the normalizers of groups of small degree very smoothly. Automorphism groups of association schemes are computed by a *backtrack method* and so are normalizers. WreathProducts are obtained from some typical association schemes. This time, we only use the lemma with some heuristics and do not use association schemes. Here the heuristics arise from the subgroups $I^{\{0..i\}} \cap K$, $i = 0, 1, \cdots, r$ in the lemma, since any of them can be used to compute the normalizer. We wrote a program of about 50 lines modifying the GAP4 Normalizer function.

2 Computing Data of Normalizers of Groups of Small Degree

Consulting on the GAP library, in 2000 small degree means ≤ 22. Now it means ≤ 30. There are 36620 transitive groups of degree $n = |\Omega|$ form 20 to 30 in the GAP4 library. We have computed the normalizers of these groups in the symmetric groups. In the following tables GAP4 means GAP4 Normalizer in $Sym(n)$ and $Alt(n)$, ISSAC2000′ revised version of the program in [3] and AC2005 the program of this talk. Table 1 shows how the programs are improved particularly in time consuming cases. Table 2 is the cumulative frequency table of Table 1 and particularly shows how the easy cases are computed. Table 3 shows some examples such that normalizers are easily computed by one program but not easily computed by another program. So from Table 1 we can see that the programs are improved in general but from Table 3 that there does not exist the best program for all groups. Some examples of hard cases for GAP are shown in the third part of Table 3. In fact there exist 755 transitive groups of degree up to 30 such that the normalizer of each group in the symmetric group was not computed within 10 hours by GAP and we stopped computing in 10 hours.

Table 1: Computing times of the Normalizers of Transitive Groups of degree n in $Sym(n)$, $20 \le n \le 30$

time	GAP4	ISSAC2000′	AC2005
$* \le$0.1sec.	10510	1829	305
0.1sec.$< * \le$0.2sec.	11728	7231	1223
0.2sec.$< * \le$0.5sec	5433	22898	12248
0.5sec.$< * \le$1sec.	2200	2973	16089
1sec.$< * \le$2sec.	1098	629	3742
2sec.$< * \le$5sec.	1015	363	1921
5sec.$< * \le$10sec.	621	182	682
10sec.$< * \le$30sec.	834	232	229
30sec.$< * \le$1min.	381	126	71
1min.$< * \le$2min.	480	40	43
2min.$< * \le$5min.	486	30	32
5min.$< * \le$10min.	357	6	8
10min.$< * \le$30min.	348	9	16
30min.$< * \le$1h.	114	12	1
1h.$< * \le$2h.	63	15	2
2h.$< * \le$5h.	112	24	3
5h.$< * \le$10h.	85	7	4
10h.$< *$	755	14	1

Table 2: Computing times of the Normalizers of Transitive Groups of degree n in $Sym(n)$, $20 \le n \le 30$ (cumulative frequency)

time	GAP4	ISSAC2000′	AC2005
$* \le$0.1sec.	10510	1829	305
$* \le$0.2sec.	22238	9060	1528
$* \le$0.5sec.	27671	31958	13776
$* \le$1sec.	29871	34931	29865
$* \le$2sec.	30969	35560	33607
$* \le$5sec.	31984	35923	35528
$* \le$10sec.	32605	36105	36210
$* \le$30sec.	33439	36337	36439
$* \le$1min.	33820	36463	36510
$* \le$2min.	34300	36503	36553
$* \le$5min.	34786	36533	36585
$* \le$10min.	35143	36539	36593
$* \le$30min.	35491	36548	36609
$* \le$1h.	35605	36560	36610
$* \le$2h.	35668	36575	36612
$* \le$5h.	35780	36599	36615
$* \le$10h.	35865	36606	36619
$-$	36620	36620	36620

Table 3: Computing times of the Normalizers of some TransitiveGroup(n,k) in $Sym(n)$ (in seconds)

n	k	GAP4	ISSAC2000′	AC2005
27	388	57	49	4782
27	583	43	14	9686
27	620	6	16	6153
27	863	1.6	1.6	849
27	890	24	15	954
30	293	1248	286	47279
30	300	98	110	34505
30	545	82	35	19726
30	563	215	57	19629
30	826	9	7	12504
30	840	58	52	7949
24	2930	2	10	104
28	1821	0.05	1.79	0.91
24	24924	1852	5938	0.3
28	1075	16	72	0.7
30	1149	28	88	1.6
30	1367	39	122	0.4
20	888	8137	0.1	0.3
21	24	30390	0.2	0.2
22	15	>36000	0.3	0.1
24	2950	>36000	46	0.2
24	20417	17567	185	0.2
25	34	>36000	0.2	0.2
25	36	18393	0.2	0.2
26	29	>36000	0.5	0.2
26	76	11016	0.4	2.9
27	805	>36000	0.4	0.1
27	806	23223	1.1	0.3

References

[1] The GAP Groups: GAP - groups, algorithms and programming, version 4. *Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany and School of Mathematical and Computational Sciences, Univ. St. Andrews, Scotland*, 2000.

[2] I. Miyamoto: Computation of normalizers in symmetric groups using association scheme ftp://tnt.math.metro-u.ac.jp/pub/ac97/PROCEEDINGS/miyamoto/ (in Japanese)

[3] I. Miyamoto: Computing normalizers of permutation groups efficiently using isomorphisms of association schemes. In *Proceedings of the 2000 Inter-*

national Symposium on Symbolic and Algebraic Computation, pp 220–224, C. Traverso, ed. ACM, 2000.

[4] I. Miyamoto: Computing isomorphisms of association schemes and its applications. *J. Symbolic Comp.*, 32:133–141, 2001.