The Todd-Coxeter Algorithm

David Allemang and Jacob Raymond

December 4th, 2019

The Purpose of Todd-Coxeter

Purpose

The purpose of the Todd-Coxeter Algorithm is to enumerate the cosets of a subgroup of a group from presentations.

Ex. Enumerating the cosets of $\langle a \mid a^2 = e \rangle$ in the group presented by $\langle a, b \mid a^2 = b^2 = (ab)^4 = e \rangle$.

- Can be used to solve the Word Problem
- Can be used to generate certain polyhedron

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle$, $H = \langle e \rangle \leq G$

Coset Table

	a	b
0		

Relation Table(s)

	a	b	a	b	a	b
0						0

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle$, $H = \langle e \rangle \leq G$

Coset Table

	a	b
0	1	
1	0	

Relation Table(s)

	a	b	a	b	a	b
0	1					0
1	0					1

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle,\ H = \langle e \rangle \leq G$

Coset Table

	a	b
0	1	2
1	0	
2		0

Relation Table(s)

	a	b	a	b	a	b
0	1				2	0
1	0	2				1
2				1	0	2

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle$, $H = \langle e \rangle \leq G$

Coset Table

	a	b
0	1	2
1	0	3
2		0
3		1

Relation Table(s)

	a	b	a	b	a	b
0	1	3			2	0
1	0	2			3	1
2			3	1	0	2
3			2	0	1	3

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle$, $H = \langle e \rangle \leq G$

Coset Table

	a	b
0	1	2
1	0	3
2	4	0
3		1
4	2	

Relation Table(s)

	a	b	a	b	a	b
0	1	3		4	2	0
1	0	2	4		3	1
2	4		3	1	0	2
3		4	2	0	1	3
4	2	0	1	3		4

Description of Todd-Coxeter

Ex. $G = \langle a, b \mid a^2 = b^2 = (ab)^3 \rangle,\ H = \langle e \rangle \leq G$

Coset Table

	a	b
0	1	2
1	0	3
2	4	0
3	5	1
4	2	$\underline{5}$
5	3	$\underline{4}$

Relation Table(s)

	a	b	a	b	a	b
0	1	3	$\underline{5}$	$\underline{4}$	2	0
1	0	2	$\underline{4}$	$\underline{5}$	3	1
2	4	5	3	1	0	2
3	5	4	2	0	1	3
4	2	0	1	3	5	4
5	3	1	0	2	4	5

Assumptions

- Only dealing with Coxeter Groups (and subgroups which are Coxeter Groups)
- Generators are their own inverses (always the case with Coxeter groups)
- "Coincidences" do not occur

Basic CPU Model

Memory transactions

Using a naïve algorithm, the following number of memory transactions are needed:

	Coset Table	Relation Table	Relations
Read	$C\sum_i [\lvert r_i\rvert - 1]$	$C\sum_i [\lvert r_i\rvert - 1]$	$C\sum_i [\lvert r_i\rvert - 1]$
Write	$C\binom{g}{2}$	$C\sum_i [\lvert r_i\rvert + 1]$	N/A (part of input)

For a total of

$$C\sum_i [4|r_i| - 1] = C\left[4\sum_i |r_i| - \binom{g}{2}\right]$$

memory transactions

Specific Groups

For the specific groups we tested on CPU, these are the amount of memory transactions needed:

Group	Transactions
T_n	$8n^2(4n+13)$
B_n	$2^{n-1}n!(7n^2+n)$
H_4	$892,800$
E_6	$6,480,000$
E_7	$496,419,840$

Expected CPU times

Assuming a 100 GB/s capacity for a Mamba Node (each transaction is 4 bytes), 35ns latency per transaction.

Group	Transactions	Time (s)
T_{50}	$4,260,000$	0.32
T_{100}	$33,040,000$	2.48
T_{150}	$110,340,000$	8.28
T_{200}	$260,160,000$	19.51
T_{250}	$506,500,000$	38.00
B_5	$345,600$	0.026
H_4	$892,800$	0.067
B_6	$5,944,320$	0.45
E_6	$6,480,000$	0.49
B_7	$112,896,000$	8.47
E_7	$496,419,840$	37.23
B_8	$2,353,397,760$	176.51

Basic CPU implementation

Pretty much exactly like the demonstration, parallelized relation table handling. 24 Hour total runtime

Group	Exp. Time (s)	Act. Time (s)
T_{50}	0.32	0.85
T_{100}	2.48	13.44
T_{150}	8.28	76.21 (1m16s)
T_{200}	19.51	331.47 (5m30s)
T_{250}	38.00	1312.11 (21m50s)
B_5	0.026	0.11
H_4	0.067	0.59
B_6	0.45	9.96
E_6	0.49	11.15
B_7	8.47	8472.46 (2h21m12s)
E_7	37.23	N/A (>21h)
B_8	176.51	N/A

Basic CPU cos/s

Group	Exp. cos/s	Act. cos/s
T_{50}	$31,250$	$11,764$
T_{100}	$16,129$	$2,976$
T_{150}	$10,869$	$1,180$
T_{200}	$8,200$	482
T_{250}	$8,333$	190
B_5	$147,692$	$34,909$
H_4	$214,925$	$24,406$
B_6	$102,400$	$4,626$
E_6	$105,795$	$4,649$
B_7	$76,165$	76
E_7	$77,975$	N/A
B_8	$58,477$	N/A

Optimized CPU Model

Optimizations

- Remove rows from Relation Table once complete
- Remove rows from Relation Table if coset appears in previous row
- Transpose Relation Tables
  - Only store start and end index/coset
- Store relations as 2 values, and use modulus on index to get generator, instead of lookup
- Add a coset after a single pass through the relation tables

Updated CPU Model

With some algorithmic optimization and carefully considering implementations details, the following amounts of memory transactions are required:

	Coset Table	Relation Table	Relations
Read	$\frac{C}{2}\sum_i \frac{C}{\lvert r_i \rvert}$	$\frac{C}{2}\sum_i \frac{C}{\lvert r_i \rvert}$	$2Cg$
Write	Cg	Cg	N/A (part of input)

For a total amount of memory transactions being

$$C\left[4g + C\sum_i \lvert r_i \rvert^{-1}\right]$$

Updated Expected Transactions

Updated expected memory transactions

Group	Basic	Updated
T_n	$8n^2(4n+13)$	$32n^2(n^2+n+2)$
B_n	$2^{n-1}n!(7n^2+n)$	$2^n n!\left[4n+2^n n!\left(\frac{3n^2+13n-5}{12}\right)\right]$
H_4	$892,800$	$490,982,400$
E_6	$6,480,000$	$17,917,148,160$
E_7	$496,419,840$	$80,062,673,080,320$

Expected CPU times

Assuming a 100 GB/s capacity for a Mamba Node (each transaction is 4 bytes).

Group	Transactions	Time (s)
T_{50}	$204,160,000$	0.0082
T_{100}	$3,232,640,000$	0.13
T_{150}	$16,309,440,000$	0.65
T_{200}	$51,458,560,000$	2.06
T_{250}	$125,504,000,000$	5.02
B_5	$165,964,800$	0.0066
H_4	$490,982,400$	0.020
B_6	$32,028,549,120$	1.28
E_6	$17,917,148,160$	0.72
B_7	$8,080,842,792,960$	323.23 (5m22s)
E_7	$80,062,673,080,320$	3202.51 (53m22s)
B_8	$2,583,644,618,096,640$	103346.0 (1d4h42m)

Optimized CPU Implementation

Group	Basic Time (s)	Exp. Time (s)	Act. Time (s)
T_{50}	0.85	0.0082	0.15
T_{100}	13.44	0.13	0.63
T_{150}	76.21	0.65	1.45
T_{200}	331.47	2.06	2.76
T_{250}	1312.11	5.02	4.88
B_5	0.11	0.0066	0.063
H_4	0.59	0.020	0.24
B_6	9.96	1.28	1.26
E_6	11.15	0.72	1.49
B_7	8472.46	323.23	148.37 (2m18s)
E_7	N/A	3202.51	2938.87 (48m58s)
B_8	N/A	103346.0	N/A

Optimized CPU cos/s

Group	Basic cos/s	Exp. cos/s	Act. cos/s
T_{50}	$11,764$	$1,219,512$	$66,666$
T_{100}	$2,976$	$307,692$	$63,492$
T_{150}	$1,180$	$138,461$	$62,068$
T_{200}	482	$77,669$	$57,971$
T_{250}	190	$49,800$	$51,229$
B_5	$34,909$	$581,818$	$60,952$
H_4	$24,406$	$720,000$	$60,000$
B_6	$4,626$	$36,000$	$36,571$
E_6	$4,649$	$72,000$	$34,791$
B_7	76	$1,995$	$4,348$
E_7	N/A	906	987
B_8	N/A	99	N/A

Comparison to GAP

Groups, Algorithms, Programming (GAP) - a System for Computational Discrete Algebra gap-system.org

Contains an implementation of Todd-Coxeter

Group	Ours (s)	Gap (s)
H_4	0.24	0.184
E_6	1.49	0.942
B_7	148.37	14.122

Basic GPU Model

Basic GPU

Naïvely we should expect similar memory accesses as CPU.

- Flatten all tables
- Lots of parallelism in relation tables
  - Execute "solving" passes on GPU
  - Race conditions still don't matter.
  - Checking when complete is expensive
- No parallelism in adding cosets
  - Major synchronization bottleneck
  - Either execute on device, or copy to host

Expected GPU times

Assuming a 240 GB/s capacity (from NVIDIA) and 10 ns latency per transaction for a Mamba Node (1 K80).

Group	Transactions	Time (s)
T_{50}	$4,260,000$	0.043
T_{100}	$33,040,000$	0.33
T_{150}	$110,340,000$	1.11
T_{200}	$260,160,000$	2.61
T_{250}	$506,500,000$	5.073
B_5	$345,600$	0.0035
H_4	$892,800$	0.0089
B_6	$5,944,320$	0.060
E_6	$6,480,000$	0.065
B_7	$112,896,000$	1.13
E_7	$496,419,840$	4.97
B_8	$2,353,397,760$	23.57

Actual GPU times

Group	Exp. Time (s)	Act. Time (s)
T_{50}	0.043	3.89485
T_{100}	0.33	22.3504
T_{150}	1.11	107.5
T_{200}	2.61	256.386
T_{250}	5.073	508.957
B_5	0.0035	1.24269
H_4	0.0089	4.30961
B_6	0.060	20.8581
E_6	0.065	23.1169
B_7	1.13	1382.62
E_7	4.97	14806.1
B_8	23.57	N/A

Basic GPU cos/s

Group	Exp. cos/s	Act. cos/s
T_{50}	$232,558$	$2,567$
T_{100}	$121,212$	$1,789$
T_{150}	$81,081$	837
T_{200}	$61,302$	624
T_{250}	$49,280$	491
B_5	$1,097,142$	$3,090$
H_4	$1,617,977$	$3,341$
B_6	$768,000$	$2,209$
E_6	$797,538$	$2,242$
B_7	$570,902$	466
E_7	$584,112$	196
B_8	$437,926$	N/A

Optimized GPU Model

Optimized GPU

- Similar algorithm optimizations as on CPU
  - Compact relations
  - Don't check for completeness (Coxeter groups only)
  - "Forward deletion" not possible
- Store relation data in constant memory
- Pre-fetch coset data for new coset scanning

Improved Model

Considering algorithm optimizations and restrictions (forward deletion):

	Coset Table	Relation Table	Relations
Read	$\frac{C^2}{2}\sum_i \lvert r_i \rvert$	$\frac{C^2}{2}\sum_i \lvert r_i \rvert$	$\frac{C^2}{2}\sum_i \lvert r_i \rvert$
Write	$C\binom{g}{2}$	$C\sum_i \lvert r_i \rvert$	N/A (part of input)

For a total memory transactions being

$$C\binom{g}{2} + \left(\frac{3}{2}C^2 + C\right)\sum_i |r_i|$$

PCIe cost for coset table scanning: $C(\frac{Cg}{4 \cdot 2.5Gtps} + 7.5 \cdot 10^{-6}s)$

Updated Expected Transactions

Group	Basic	Updated
T_n	$8n^2(4n+13)$	$8n^2(6n^3+24n^2+n+7)$
B_n	$2^{n-1}n!(7n^2+n)$	$2^n n!\left[n^2\left(\frac{3}{2}2^n n!+1\right)+\frac{n(n-1)}{2}\right]$
H_4	$892,800$	$5,288,011,200$
E_6	$6,480,000$	$141,090,336,000$
E_7	$496,418,840$	$606,790,369,704,960$

Expected GPU Times

Group	Transactions	Time (s)
T_{50}	$16,201,140,000$	16
T_{100}	$499,208,560,000$	501
T_{150}	$3,742,228,260,000$	$3,757$ (1h)
T_{200}	$15,667,266,240,000$	$15,732$ (4h)
T_{250}	$47,625,128,500,000$	$47,823$ (13h)
B_5	$553,094,400$	0.5554
H_4	$5,288,011,200$	5.3100
B_6	$114,664,135,680$	115
E_6	$141,080,336,000$	141
B_7	$30,589,261,516,800$	$30,716$ (8.5h)
E_7	$606,790,369,704,960$	$609,318$ (7d)
B_8	$10,228,036,068,311,040$	$10,270,652$ (118d)

Optimized GPU Times

Group	Basic Time (s)	Exp. Time (s)	Act. Time (s)
T_{50}	3.89485	16	4.33001
T_{100}	22.3504	501	23.661
T_{150}	107.5	$3,757$	103.069
T_{200}	256.386	$15,732$	246.577
T_{250}	508.957	$47,823$	494.363
B_5	1.24269	0.5554	1.35973
H_4	4.30961	5.3100	4.76805
B_6	20.8581	115	22.5943
E_6	23.1169	141	25.1293
B_7	1382.62	$30,716$	1350.11
E_7	14806.1	$609,318$	14657.7
B_8	N/A	$10,270,652$	N/A

Optimized GPU cos/s

Group	Basic cos/s	Exp. cos/s	Act. cos/s
T_{50}	$2,567$	625	$2,309$
T_{100}	$1,789$	79	$1,690$
T_{150}	837	23	873
T_{200}	624	10	648
T_{250}	491	5	505
B_5	$3,090$	$6,913$	$2,824$
H_4	$3,341$	$2,711$	$3,020$
B_6	$2,209$	400	$2,039$
E_6	$2,242$	367	$2,062$
B_7	466	21	477
E_7	196	4	198
B_8	N/A	1	N/A