Word posets, with applications to Coxeter groups

Matthew J. Samuel

Rutgers University—New Brunswick

WORDS 2011

Introduction

- "Word posets" are partially ordered sets that capture the structure of commutation classes of words in monoids
- Allow one to rephrase enumeration questions in terms of reasonably well-understood statistics of partially ordered sets
- Particularly effective when applied to a special type of monoid known as a Coxeter group

Recall. . .

- A monoid generated by a set S of symbols is the set of equivalence classes of words in S of finite length (including the empty word) via a particular kind of equivalence relation
- The equivalence relation is determined by relations of the form $u = v$, where u and v are two words
- $u = v$ means that whenever u occurs as a contiguous subsequence of some word, it can be replaced with v to obtain an equivalent word
- **Example:** If $abc = da$, then $dfabcp = dfdap$

Commutation classes

- If $a, b \in S$ and $ab = ba$, then a and b are said to commute
- $ab = ba$ is called a commutation relation
- If the word w' can be obtained from w using only commutation relations, then in the same commutation class
- **Example:** If $ab = ba$, $cd = dc$, and $ad = da$, then the sequence of replacements

$$abcd \to bacd \to badc \to bdac$$

shows that $abcd$ and $bdac$ are in the same commutation class

The idea of word posets

- Again suppose $ab = ba$, $cd = dc$, and $ad = da$
- Then the commutation class of $abcd$ is $\{abcd, bacd, badc, bdac, abdc\}$
- "The set of words of length 4 using all of the symbols a, b, c, and d such that a comes before c and b comes before both c and d"
- Partial ordering on the symbols describes the commutation class

Word posets

- Ordering the symbols themselves is not sufficient in general
- Fix a word, order indices instead
- The partial ordering on $\{1, 2, 3, 4\}$ corresponding to the word *abcd* on the previous slide: $1 < 3$, $2 < 3$, $2 < 4$, all other pairs of indices incomparable
- The partially ordered set with 4 elements

$$\{1, 2, 3, 4\}$$

together with the labelling

$$1 \mapsto a, 2 \mapsto b, 3 \mapsto c, 4 \mapsto d,$$

form the word poset for *abcd* in this monoid

Word posets and commutation classes

- Different words in the same commutation class always give different labellings and may give different partial orderings
- *bdac* gives $1 < 2$, $1 < 4$, and $3 < 4$, which is a different partial ordering on $\{1, 2, 3, 4\}$ than for *abcd*
- However, different word posets for words in the same commutation class are isomorphic in a manner preserving the labelling

Word poset axioms

- A word poset is a pair (P, s) consisting of a finite partially ordered set P together with a function $s : P \to S$
- Must satisfy the following for all $x, y \in P$:
  - If $s(x)$ and $s(y)$ are either equal or do not commute, then either $x \leq y$ or $y \leq x$
  - If $x < y$ and there is no z such that $x < z < y$, then $s(x)$ and $s(y)$ are either equal or do not commute

Word posets characterize commutation classes

- If we have two word posets (P, s) and (P', s'), then said to be isomorphic if there is a bijective function $f : P \to P'$ such that $s'(f(x)) = s(x)$ for all $x \in P$ and for all $x, y \in P$ we have $f(x) \leq f(y)$ if and only if $x \leq y$ (poset isomorphism preserving the labelling)
- Then

Theorem

The isomorphism classes of word posets (P, s) with n elements are in one-to-one correspondence with the commutation classes of words in the monoid of length n.

Getting the words back

- If we have a word poset (P, s) and want to find the words in the commutation class, need to put P in a linear order and write the symbols in this order according to the labelling
- A linear extension of a partially ordered set P with n elements is a bijective function $f : P \to \{1, 2, \ldots, n\}$ such that $f(x) \leq f(y)$ whenever $x \leq y$ in P
- To every linear extension $f : P \to \{1, 2, \ldots, n\}$ we may associate a word $w(f)$ such that $w(f)_i = s(f^{-1}(i))$ for $1 \leq i \leq n$

Linear extensions correspond to words

Theorem

The correspondence $f \mapsto w(f)$ between linear extensions of (P, s) and words in the associated commutation class is bijective.

Counting linear extensions and words

- Word posets can be constructed in polynomial time given monoid relations and a word
- Given a partially ordered set P, can use P as an alphabet and the partial ordering to construct monoid relations so that P is the word poset for any of its linear extensions
- Thus, finding the number of linear extensions of a partially ordered set is polynomial time equivalent to finding the number of elements in a commutation class of words in some monoid

Computational complexity

- A function problem is said to be in #P if it counts the number of solutions to an NP decision problem, and a problem is said to be #P-complete if every problem in #P can be reduced to it in polynomial time
- The problem of counting linear extensions is known to be #P-complete, so it follows that

Theorem

The problem of counting the numbers of words in commutation classes of monoids is #P-complete.

Coxeter groups

- A Coxeter group is a monoid on some alphabet S satisfying the following properties:
  - $aa = a^2$ is equivalent to the empty word (which we will denote by 1) for all $a \in S$
  - The only other relations are of the form $aba \cdots = bab \cdots$, where the left and right sides are words of the same length $m > 1$ in exactly two symbols $a, b \in S$ such that no two adjacent symbols are equal
- **Example:** If $S = \{a, b, c\}$ and W is determined by $a^2 = b^2 = c^2 = 1$, $aba = bab$, $bcb = cbc$, and $ac = ca$, then W is a Coxeter group with 24 elements
- A pair (W, S), where W is a Coxeter group and S is the generating alphabet, is called a Coxeter system

Reduced words

- A word in a Coxeter group is said to be reduced if there is no equivalent word of shorter length
- **Example:** if $aba = bab$ (and $ab \neq ba$), then aba and bab are reduced words but $abab$ is not, because $(aba)b = (bab)b = ba(bb) = ba$
- Coxeter groups generated by an alphabet S are characterized by the conditions
  - $a^2 = 1$ for all $a \in S$
  - If a word is not reduced, then some pair of symbols can be deleted to obtain an equivalent word

Counting reduced words

- If the length of a reduced word is n, then there are at most $n!$ equivalent reduced words (this is not obvious)
- In particular, an equivalence class of words in a Coxeter group contains only finitely many reduced words
- Reducing counting linear extensions of a partially ordered set to counting words in a commutation class of a monoid can be done in a Coxeter group whose only relations are commutation relations
- Recognizing reduced words for an element can be done in polynomial time, so counting reduced words is in #P

Complexity of counting reduced words

- By the remarks on the previous slide, it follows that

Theorem

The problem of counting reduced words of elements of Coxeter groups is #P-complete.

- The best algorithm I know of runs in $O(n2^n)$, where n is the length of the word
- I do not know how to reduce counting reduced words to counting linear extensions in polynomial time

Formula for counting reduced words

- An element w in a Coxeter group must have finitely many commutation classes of reduced words
- Let $\mathrm{WP}(w)$ denote the set of word posets corresponding to commutation classes of reduced words for w
- If P is a partially ordered set, let $E(P)$ denote the number of linear extensions of P
- Then the number of reduced words for w is

$$\sum_{(P,s)\in\mathrm{WP}(w)} E(P)$$

Finding the word posets

- Let $\ell(w)$ denote the length of a reduced word for w
- Recursive method for constructing all reduced word posets for w:
  (1) If $\ell(w) = 0$, then the only reduced word poset for w is the empty partially ordered set with its unique labelling
  (2) If $a \in S$ is such that $\ell(aw) < \ell(w)$ and $(P, s) \in \mathrm{WP}(aw)$, then adjoin a new element x to P with $s(x) = a$ that is less than a given minimal element $y \in P$ if and only if a and $s(y)$ do not commute to obtain $(P \cup \{x\}, s) \in \mathrm{WP}(w)$
  (3) Iterate (2) over all $a \in S$ such that $\ell(aw) < \ell(w)$, at least one of which must exist if $\ell(w) > 0$

Counting the word posets

- Let $C(w) = |\mathrm{WP}(w)|$ denote the number of reduced word posets for w, or equivalently the number of commutation classes of reduced words
- Explicit construction on previous slide leads to a formula for $C(w)$:

$$C(w) = \sum (-1)^{|T|+1} C(Tw)$$

  where T ranges over all nonempty subsets of S such that
  - all elements of T commute with each other,
  - $\ell(aw) < \ell(w)$ for all $a \in T$,

  and Tw denotes the product of w with all of the elements of T on the left

Bounding $C(w)$

- We have the following bound.

Theorem

For any element w of a Coxeter group such that $\ell(w) > 0$,

$$C(w) \leq \frac{2}{3} 3^{\frac{1}{2}\ell(w)}.$$

- I conjecture that if α is a constant, $f(n)$ is a function such that $\lim_{n \to \infty} \frac{\log f(n)}{n} = 0$, and

$$C(w) \leq f(\ell(w)) \alpha^{\ell(w)}$$

for all w, then $\alpha \geq 3^{\frac{1}{2}} \approx 1.732$. Direct computation has shown that at least $\alpha > 1.715$

Primitive sorting networks

- The number $P(n)$ of primitive sorting networks on n elements is the same as the number of commutation classes of reduced words for a particular element of length $N_n = \frac{n(n-1)}{2}$
- Thus, $P(n) \leq \frac{2}{3} 3^{\frac{1}{2} N_n}$, and I can show that $P(n) \geq P(m)^{\frac{N_n}{N_m}}$ for all m and infinitely many $n > m$
- It was previously known that

$$0.23105 \approx \frac{1}{3} \log 2 \leq \lim_{n \to \infty} \frac{\log P(n)}{N_n} \leq \log 2 \approx 0.69315$$

and the facts above together with the computation of $P(12)$ show that

$$0.53941 \approx \frac{1}{66} \log P(12) \leq \lim_{n \to \infty} \frac{\log P(n)}{N_n} \leq \frac{1}{2} \log 3 \approx 0.54931.$$

Conjecture as to the limit

- These elements seem to have the most commutation classes by their length, so I conjecture that

Conjecture

We have

$$\lim_{n\to\infty} \frac{\log P(n)}{N_n} = \frac{1}{2}\log 3.$$