Transposition Switches in Series

M. H. Albert[1] R. E. L. Aldred[2] M. D. Atkinson[1] H. P. van Ditmarsch[1] C. C. Handley[1] D. A. Holton[2] D. J. McCaughan[2]

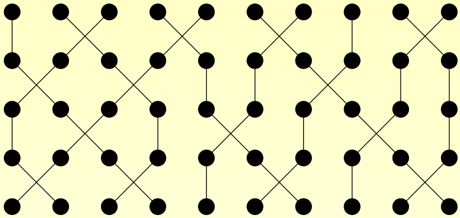
[1]Department of Computer Science
University of Otago

[2]Department of Mathematics and Statistics
University of Otago

Taupo, Dec 2004

Outline

- Definitions
  - Permuting Machines
  - Permutation Classes
- Two Simple Machines
- Conclusions



Permuting Machines

- A *permuting machine* is an (abstract) device whose only function is to accept a stream of input tokens and produce from them a stream of output tokens that forms a permutation of the input.
- The *names* of the input items are unimportant (they are featureless boxes).
- If some input items are omitted the machine is capable of rearranging the remaining ones as they were rearranged in the original.
- Standing example: A stack.

Involvement

- Permutations are written as sequences of values.
- Given two permutations $\sigma \in \mathcal{S}_k$ and $\pi \in \mathcal{S}_n$ we say that σ *is involved in* π and write $\sigma \preceq \pi$ if there is a subsequence of π whose elements occur in the same relative order as σ.
- If σ is not involved in π we say that π *avoids* σ.
- For example $312 \preceq 253146$ due to the sequences 514 or 534.

Permutation Classes

- A *permutation class*, $\mathcal{C}$, is a set of permutations closed downwards in the involvement order (i.e. $\pi \in \mathcal{C}$ and $\sigma \preceq \pi$) implies $\sigma \in \mathcal{C}$).
- A permutation class, $\mathcal{C}$, can also be characterized as the collection of permutations avoiding some set B of permutations. When B is the set of minimal (with respect to $\preceq$) permutations not in $\mathcal{C}$ it is called the *basis* of $\mathcal{C}$.
- The *enumeration sequence*

$$e(\mathcal{C}) := (|\mathcal{C} \cap \mathcal{S}_n|)_{n=0}^{\infty}.$$

A stack (after Knuth)

- The set of permutations, $\mathcal{K}$, produced by a stack all avoid 312
- Conversely if π avoids 312 it can be produced (essentially uniquely) by a stack.
- The enumeration sequence:

$$e(\mathcal{K}) = \left(1, 2, 5, 14, 42, \ldots, \frac{1}{n+1}\binom{2n}{n}, \ldots\right)$$

- A structural decomposition:

$$\mathcal{K} = () + (\mathcal{K} \ominus \{1\}) \oplus \mathcal{K}.$$

- All these observations make $\mathcal{K}$ the prototypical exemplar of a *tractable* class.

The Big Questions

Given a class $\mathcal{C}$:

- What is its basis?
- What is its enumeration sequence?
- How easily can we recognize elements of $\mathcal{C}$?
- Can we provide a structural characterization of the elements of $\mathcal{C}$?
- How complex is $\mathcal{C}$ as a partial order? In particular does it contain any infinite antichains?

A class is definitely tractable if we can answer all these questions satisfactorily.

Transposition Machines

Imagine a robot arm at the head of an input stream. It can pick up one or two items and place it or them on an output stream in either order. There are actually two distinct versions of this machine:

Buffer Arm When holding two items, the arm may place either one or both of them on the output stream.

Transposition Arm When holding two items, the arm *must* place both on the output stream.

Let the class associated to the first machine be called $\mathcal{B}$ and that associated to the second $\mathcal{T}$.

Both $\mathcal{B}$ and $\mathcal{T}$ Are (Very) Tractable

- The basis of $\mathcal{B}$ is $\{312, 321\}$. Its enumeration sequence is:

  $$e(\mathcal{B}) = (1, 2, 4, \ldots, 2^{n-1}, \ldots).$$

  The elements of $\mathcal{B}$ are recognized as "those permutations where each successive element is either the smallest or second smallest of the remaining elements". A number of structural characterizations of $\mathcal{B}$ can be found.
- The basis of $\mathcal{T}$ is $\{231, 312, 321\}$. Its enumeration sequence is:

  $$e(\mathcal{T}) = (1, 2, 3, 5, 8, 13, 21, \ldots).$$

  Structurally:

  $$\mathcal{T} = () + \{1, 21\} \oplus \mathcal{T}.$$

  and this also provides a recognition algorithm.

Compositions of $\mathcal{B}$

The k-fold composition $\mathcal{B}^k$ is almost as simple as $\mathcal{B}$.

- The spare hand on each of the first $k-1$ robot arms, plus both hands of the last can be used as a buffer.
- The permutations in $\mathcal{B}^k$ are precisely those where each element is among the $(k+1)$ smallest of the remaining elements.

$$e(\mathcal{B}^k) = (1, 2, 6, \ldots, k!, (k+1)k!, (k+1)^2k!, \ldots)$$

- The basis consists of all the permutations in $\mathcal{S}_{k+2}$ that begin with $k+2$.
- Note that we also have a simple algorithm which, given $\pi \in \mathcal{S}$ determines the minimum k such that $\pi \in \mathcal{B}^k$.

Compositions of $\mathcal{T}$

The structure of $\mathcal{T}^k$ seems to be a little more complex. However, at least for fixed k we do have some degree of tractability.

Theorem

Let k be a fixed positive integer. Then there is a natural bijection between $\mathcal{T}^k$ and a regular language. Moreover, there exists a linear time recognition algorithm for $\mathcal{T}^k$, and for its basis.

Using standard techniques this theorem also shows that the enumeration sequence $e(\mathcal{T}^k)$ satisfies a linear recurrence with constant coefficients.

Ideas of the Proof

- The permuting machine which defines $\mathcal{T}$ is much like a finite state machine.
- This analogy can be pushed to show that, with respect to the encoding of permutations given by replacing each element of a permutation by its rank among the remaining elements, $\mathcal{T}$ is a regular language.
- Moreover, composition of *regular permuting machines* preserves regularity.

But . . .

- For $k \geq 5$, $\mathcal{T}^k$ does not have a finite basis. Show me!
- The generating functions become relatively complex:

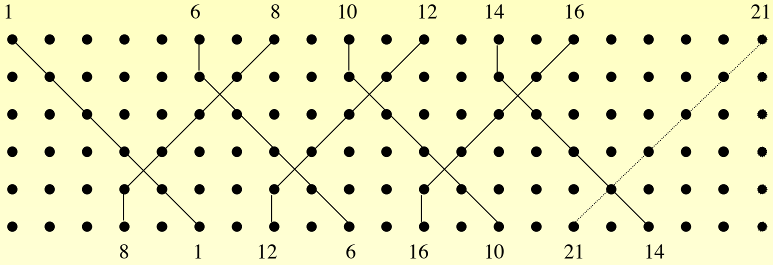
$$\sum e(\mathcal{T}^2)_n x^n = \frac{1-x}{1-2x-x^3}$$

$$\sum e(\mathcal{T}^3)_n x^n = \frac{1-x-2x^2+x^3-x^4}{1-2x-2x^2-6x^4+2x^5-x^6}.$$

- We have no algorithm which, given π, finds the minimum k such that $\pi \in \mathcal{T}^k$.

Conclusions

- Composition of permutation classes introduces complex structure.
- For *regular* classes, composition is well-behaved.
- **Open problem**: Determine (efficiently) the minimum number of passes required by a transposition arm in order to sort (or generate) a given permutation.



Back