Revolving door ordering

Contents

1 Minimal Change Ordering for subsets 1
  1.1 Subsets of an \( n \)-set .......................... 1
  1.2 Generating \( k \)-subsets .......................... 1
  1.3 Proofs ........................................... 2

1 Minimal Change Ordering for subsets

1.1 Subsets of an \( n \)-set

Fix \( n \), and consider the class of subsets of an \( n \)-set. We want to generate all elements of this class with a minimum change. For example, between consecutive elements in a listing, we perhaps there is a difference of a single element.

We have already explored binary strings, and we have also already explored the connection between generating binary strings and generating a subset. A Gray code for binary strings implicitly describes a minimal change exhaustive generation scheme for the set of subsets of an \( n \)-set.

Well, that was easy! Now let us make it harder.

1.2 Generating \( k \)-subsets

Let us consider a restriction: all possible \( \binom{n}{k} \) \( k \)-subsets of an \( n \)-set. We can formulate this in terms of binary strings: the difference between two binary strings must be at least two, and ideally we would like to describe a scheme in which the difference is exactly 2 bits.

First, we describe a natural minimal change order which is analogous to RBC \( R(n) \). Let us use some information that we have at our disposal:

\[
\binom{n}{0} = \binom{n}{n} = 1, \quad \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, 1 \leq k \leq n.
\]

This identity actually suggests a minimal change order for \( k \)-subsets, defined in a recursive way. RBC can be viewed as an interpretation of the identity

\[
2^n = 2^{n-1} + 2^{n-1},
\]

and we let this guide us. The number of empty subsets of an \( n \)-set is one, the empty set and this is represented

\[
R_0(n) = \underbrace{0 \ldots 0}_k.
\]

Likewise,

\[
R_0(n) = \underbrace{1 \ldots 1}_k
\]

Now, assume we have a minimal change sequence \( R_k(n - 1) \) for all \( k \). Then

\[
R_k(n) = 0 \cdot \{ R_k(n - 1) \}, 1 \cdot \{ R_{k-1}(n - 1)^R \}
\]

where \( R \) means to reverse the order of the sequence.

If we interpret the binary strings occurring in \( R_k(n) \) as \( k \)-subsets with the first bit representing the appearance or nonappearance of \( n \), the second bit \( n - 1 \) and so on, then we can rephrase this definition as follows

\[
R_k(n) = [R_k(n - 1), \{ n \} \cup R_{k-1}(n - 1)^R]
\]
The rows of \( R_k(n) \) represent a cyclic minimal change ordering of all \( k \)-subsets of \([n]\).

**Proof.** We prove this by induction on \( n \). Again \( n = 1 \) gives \( k = 1 \) and so can be observed directly.

Assume the result holds for \( n = m \geq 1 \), and for any \( 0 \leq k \leq m \). Now consider \( n = m + 1 \).

If \( k = n \) then the result can again be observed directly. Assume \( 1 \leq k \leq n - 1 \). Then the first row of \( R_k(n) \) is the first row of \( R_k(n-1) \) with an extra \( 0 \) at the beginning which by induction is as it should be. The last row of \( R_k(n) \) is the first row of \( R_{k-1}(n-1) \) with an extra \( 1 \) at the beginning, which by induction is as it should be.

Consider the distance between adjacent rows of \( R_k(n) \). The adjacent differences among the first \( \binom{n}{k} \) rows are the same as the adjacent differences between the rows of \( R_k(n-1) \) and so by induction are all \( 2 \). Likewise the adjacent differences among the last \( n - 1 - k \) rows are the same as the adjacent differences between the rows of \( R_{k-1}(n-1) \) and so by induction are all \( 2 \). It remains to check the difference between the \( \binom{n-1}{k} \)th row and its successor, and between the first and last rows.

The \( \binom{n-1}{k} \)th row of \( R_k(n) \) is the \( \binom{n-1}{k} \)th row of \( R_k(n-1) \) with an extra \( 0 \) at the beginning. The next row of \( R_k(n) \) is the \( \binom{n-1}{k-1} \)th row of \( R_{k-1}(n-1) \) with an extra \( 1 \) at the beginning. By the proposition these two rows of \( R_k(n) \) are the following two rows

\[
\begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\
1 & 1 & 0 & \cdots & 0 & 0 & 1 & \cdots & 1
\end{bmatrix}
\]

which are distance 2 apart.

The first row of \( R_k(n) \) is the first row of \( R_k(n-1) \) with an extra \( 0 \) at the beginning. The last row of \( R_k(n) \) is the first row of \( R_{k-1}(n-1) \) with an extra \( 1 \) at the beginning. By the previous proposition these rows are

\[
\begin{bmatrix}
0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 0 & 0 & 1 & \cdots & 1
\end{bmatrix}
\]

which are distance 2 apart. This completes the proof.

This generation scheme has a major drawback—To compute \( R_k(n) \) you need to compute \( R_{k-1}(n-1) \), \( R_k(n-1) \) and thus also \( R_{k-2}(n-2) \), \( R_{k-2}(n-1) \), \( R_{k-1}(n-2) \), \( R_k(n-2) \)… which will either represent a lot of repeated calculation, or a lot of storage.

Next time we’ll look at a successor function for this ordering, so that we never need to generate it by the definition.