**Gray Codes and Universal Cycles**

(Chair/Président: **Joe Sawada** (University of Guelph))

Org: **Joe Sawada** (University of Guelph) and/et **Aaron Williams** (McGill University))

---

**JOE SAWADA**, University of Guelph

*An overview of Combinatorial Generation*

This talk will provide an overview for the session, providing a brief history, general definitions, and several applications. Several recent Gray code results will be mentioned with their relationships to universal cycles.

**XI SISI SHEN**, McGill University

*A "Hot Potato" transposition Gray code for permutations*

We prove the existence of a "Hot Potato" Gray code for permutations that always transposes the value 1 with a value that is one or two positions to its left or right, circularly. This appears to be the first permutation Gray code that restricts both the values and positions of the transposed symbols. Our construction relies on the doubly-adjacent Gray code by Compton and Williamson. This is joint work with Aaron Williams.

**RYUHEI UEHARA**, Japan Advanced Institute of Science and Technology

*On generation of graphs with geometric representations*

Random generation and enumeration on a graph class have been required to test graph algorithms. In particular, unlabeled simple graphs are standard as a set of input. To enumerate unlabeled graphs, we have to avoid to generate two or more isomorphic graphs. In this talk, we summarize two recent algorithms for proper interval graphs and bipartite permutation graphs. Both of graph classes have natural intersection model of unit intervals and lines, respectively. We use the properties of the intersection models, and propose the random generation and enumeration algorithms.

**AARON WILLIAMS**, McGill University

*Iterative Gray Codes*

Gray codes are typically defined recursively. For example, the binary reflected Gray code for $n$-bit binary strings is typically expressed as $\text{BRGC}(n) = 0 \cdot \text{BRGC}(n-1), 1 \cdot \text{reflect}(\text{BRGC}(n-1))$.

We instead consider iterative definitions, where objects are created one at a time. We give a simple greedy algorithm that recreates well-known Gray codes for binary strings, permutations, combinations, binary trees, and set partitions. We also discuss successor rules, including a memoryless rule that visits vertices in the directed Cayley graph for $S_n$ generated by $\sigma = (1 \ 2 \ \cdots \ n)$ and $\tau = (1 \ 2)$.