Exercise 7: Spades of Gray

Task 1: Reflecting on Gray code

a) Given a B-bit Gray code, generate a $(B+1)$-bit Gray code by using the code twice, once forward with 0 as prefix and then reversed with 1 as prefix. Using this scheme recursively, write down a 3-bit *Binary Reflected Gray Code (BRGC).*

b) Expand the code to allow for metastability, by adding the "codewords" $G(x)*G(x+1)$ for $x \in [2^B]$, where G denotes the B-bit BRGCencoding function. We call the union of such strings and stable codewords *valid strings.* Order the valid strings such that the unstable strings stabilize only to adjacent stable codewords, where the order of the stable codewords is given by the natural order on the encoded values. Write the result down for your 3-bit code.

c) Define

$$\max_G\{G(x), G(y)\} := G(\max\{x, y\})$$
$$\min_G\{G(x), G(y)\} := G(\min\{x, y\})\,,$$

i.e., according to the order on stable BRGCstrings. Show that the metastable closure of $\max_G$ and $\min_G$ restricted to inputs that are valid strings is identical to what you would get if you extended $\max_G$ and $\min_G$ to valid strings $G(x)*G(x+1)$ according to the above order.

d) Prove that there are comparator circuits for valid strings w.r.t. the above order.

Task 2: Conversion

a) Given circuits that convert B-bit BCRG and the reversed code to unary code, respectively, provide a circuit with $\mathcal{O}(2^B)$ additional gates that converts $(B+1)$-bit BCRG to unary code.

b) Given a circuit that converts B-bit valid strings to (2^B-1)-bit unary code (mapping $G(x)*G(x+1)$ to $1^x\mathrm{M}0^{2^B-2-x}$), provide a circuit with at most 2^B-1 additional gates and depth one larger that converts the reversed code.

c) Combine the previous results into a recursive construction that converts B-bit valid strings to (2^B-1)-bit unary code with a circuit of size $\mathcal{O}(B2^B)$ and depth $\mathcal{O}(B)$.

d) Compare this result to what you would get from applying the construction from Theorem 6.6.

e*) Take a closer look at the circuits you constructed and/or the reverse code. Can you come up with a more efficient decoding circuit? By how much does size and depth improve in your new solutions?

Task 3*: Closure

a) Learn about Huffman's implementation of the metastable closure, based on prime implicants.

b) Show that implementing comparators for valid strings with the above order using his method results in very large circuits.

c) Decode what you learned in the TA session to the others!