Gray codes

Danny Nygård Hansen

19th November 2022

If a and b are binary strings of the same length, we denote the bitwise exclusive disjunction of a and b by $a \oplus b$. We denote the concatenation of a with b either by $a \circ b$ or ab. Also, if b is a binary string, denote by $b^{\gg}$ the right logical shift of b, i.e. the string obtained by removing the rightmost bit of b and appending a 0 on the left of the result.

Let $n \in \mathbb{N}$. For a number $k \in \mathbb{N}$ with $k < 2^n$ we denote the n-bit binary representation of k by $\mathrm{bin}_n(k)$. Furthermore, we denote the n-bit Gray code for k by $\mathrm{gr}_n(k)$. By definition, $\mathrm{gr}_0(0) = \lambda$ and

$$\mathrm{gr}_{n+1}(k) = \begin{cases} 0 \circ \mathrm{gr}_n(k), & k < 2^n, \\ 1 \circ \mathrm{gr}_n(2^{n+1} - 1 - k), & k \geq 2^n. \end{cases}$$

for all $n \in \mathbb{N}$ and $(n+1)$-bit numbers k. We claim the following:

Proposition 0.1

Let $n \in \mathbb{N}$, and let $k \in \mathbb{N}$ be an n-bit number. Writing $\mathrm{bin}_n(k) = b_{n-1} \cdots b_0$ we have $\mathrm{gr}_n(k) = a_{n-1} \cdots a_0$, where $a_{n-1} = b_{n-1}$ and

$$a_i = b_{i+1} \oplus b_i \tag{0.1}$$

for $i \in \{0, \dots, n-2\}$. That is,

$$\mathrm{gr}_n(k) = b_{n-1}(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0).$$

Conversely we have

$$b_i = a_i \oplus \cdots \oplus a_{n-1}.$$

The formula (0.1) also holds in the case $i = n-1$ if we let $b_n = 0$, i.e. we prepend zeros if necessary.

PROOF. If $n = 0$, then the claim is obvious since there are no 0-bit numbers. Now let k be an $(n+1)$-bit number, so that $k < 2^{n+1}$, and write $\mathrm{bin}_{n+1}(k) = b_n \cdots b_0$. If $k < 2^n$, then $b_n = 0$ and $\mathrm{gr}_{n+1}(k) = 0 \circ \mathrm{gr}_n(k)$. By induction we have

$$\begin{aligned}\mathrm{gr}_n(k) &= b_{n-1}(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0)\\ &= (b_n \oplus b_{n-1})(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0),\end{aligned}$$

so it follows that

$$\mathrm{gr}_{n+1}(k) = b_n \circ \mathrm{gr}_n(k) = b_n(b_n \oplus b_{n-1})(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0)$$

as claimed. If instead $k \geq 2^n$, then $b_n = 1$. Writing $k = 2^n + r$ with $0 \leq r < 2^n$ we have $\mathrm{bin}_n(r) = b_{n-1} \cdots b_0$. Now notice that $\mathrm{bin}_n(2^n - 1 - r) = \bar{b}_{n-1} \cdots \bar{b}_0$ since

$$(\bar{b}_{n-1} \cdots \bar{b}_0)_2 + r + 1 = (\bar{b}_{n-1} \cdots \bar{b}_0)_2 + (b_{n-1} \cdots b_0)_2 + 1 = 2^n.$$

By induction we have

$$\begin{aligned}\mathrm{gr}_n(2^n - 1 - r) &= \bar{b}_{n-1}(\bar{b}_{n-1} \oplus \bar{b}_{n-2}) \cdots (\bar{b}_1 \oplus \bar{b}_0)\\ &= (b_n \oplus b_{n-1})(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0)\end{aligned}$$

since $b_n = 1$, so it follows that

$$\begin{aligned}\mathrm{gr}_{n+1}(k) &= b_n \circ \mathrm{gr}_n(2^n - 1 - r)\\ &= b_n(b_n \oplus b_{n-1})(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0)\end{aligned}$$

as desired.

For the final claim, simply notice that

$$\begin{aligned}a_i \oplus \cdots \oplus a_{n-1} &= (b_i \oplus b_{i+1}) \oplus (b_{i+1} \oplus b_{i+2}) \oplus \cdots \oplus (b_{n-2} \oplus b_{n-1}) \oplus b_{n-1}\\ &= b_i \oplus (b_{i+1} \oplus b_{i+1}) \oplus (b_{i+2} \oplus \cdots \oplus b_{n-2}) \oplus (b_{n-1} \oplus b_{n-1})\\ &= b_i.\end{aligned}$$

Alternatively we may notice that (0.1) defines a linear system of equations with coefficients in $\mathbb{Z}/2\mathbb{Z}$ and invert this. □

COROLLARY 0.2

For $n \in \mathbb{N}$ and any n-bit number k, we have

$$\mathrm{gr}_n(k) = \mathrm{bin}_n(k) \oplus \mathrm{bin}_n(k)^{\gg}.$$

PROOF. Writing $\mathrm{bin}_n(k) = b_{n-1} \cdots b_0$, the proposition implies that

$$\begin{aligned}\mathrm{gr}_n(k) &= b_{n-1}(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0)\\ &= (0 \oplus b_{n-1})(b_{n-1} \oplus b_{n-2}) \cdots (b_1 \oplus b_0).\end{aligned}$$

But $\mathrm{bin}_n(k)^{\gg} = 0 b_{n-1} \cdots b_1$, so the claim follows. □

References

Knuth, Donald E. (2011). *The Art of Programming, Volume 4A: Combinatorial Algorithms, Part 1*. 1st ed. Addison-Wesley. 883 pp. ISBN: 978-0-201-03804-0.