10.4.2014

Gray-Codes

Ziel: Finde eine Folge von Bitstrings (der Länge n), sodass sich nur genau ein Bit zwischen zwei aufeinanderfolgenden Strings ändern.

Def: Sei Γ ein beliebiges Alphabet, dann bezeichnet Γ_n eine Folge von Strings der Länge n bzgl. einer festgelegten Ordnung.

Sei $a \in \Gamma$ und $\Gamma_n = (w_1, w_2, \dots, w_m)$ eine Folge von Strings der Länge n über Γ, dann ist $a\Gamma_n =_{def} (aw_1, aw_2, \dots, aw_m)$ eine Folge von Strings der Länge $n+1$, deren Prefix a ist.
Weiterhin sei $\Gamma_n^R = (w_m, w_{m-1}, \dots, w_1)$ die Folge in umgekehrter Reihenfolge

Def: (Binärer) Graycode: Sei $\Gamma = \{0,1\}$, dann heißt die induktiv definierte Folge

(IA) $\Gamma_0 = \varepsilon$, wobei ε das leere Wort ist

(IS) $\Gamma_{n+1} = 0\Gamma_n \circ 1\Gamma_n^R$

(binärer) Graycode der Länge $n+1$, wenn $\circ$ die Konkatenation von Folgen von Strings bezeichnet.

<u>Bsp</u>: $n=4$

$n=1$: $(0,1)$

$n=2$: $(00, 01, 11, 10)$

$n=3$: $(000, 001, 011, 010, 110, 111, 101, 100)$

$n=4$: $(0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000)$

<u>Beo</u>: Sei $\Gamma = \{0,1\}$, dann $\Gamma_n^R = \Gamma_n \oplus 1\underbrace{0 \ldots 0}_{n-1\text{-mal}}$.

<u>Beo</u>: Sei $\Gamma = \{0,1\}$, dann gilt $\#(\Gamma_n) = 2^n$, d.h. der Graycode von n-stelligen Binärworten induziert eine Permutation auf der Menge Bitstrings. Die ersten 2^{n-1} Elemente entsprechen Γ_{n-1}.

Fasst man die Bitstrings als natürliche Zahlen auf, so kann man eine unendliche Folge Γ_∞ definieren. Sei $g : \mathbb{N} \to \bigcup_{n \geq 1} \Gamma_n$, dann

$$\Gamma_\infty = g(0), g(1), g(2), \ldots$$
$$= (0), (1), (11), (110), \ldots$$

Beo: Γ_∞ ist eine Permutation der natürlichen Zahlen und die ersten 2^n Elemente von Γ_∞ entsprechen Γ_n, wenn man die führenden Oen entfernt

Proposition: Sei $i \in \mathbb{N}$, dann ist die Hamingdistanz zwischen $g(i), g(i+1) \in \Gamma_n$ genau 1.

Beweis: Induktion über den Aufbau von Γ_n #

Lemma: Wen $k = 2^n + r$ und $0 \leq r < 2^n$, dann gilt $g(k) = \underbrace{2^n}_{\text{führende Eins}} + g\underbrace{(2^n - 1 - r)}_{\text{reflektierter Wert}}$

Beweis: Direkte Konsequenz aus der induktiven Definition und der Tatsache, dass 2^n-1-r der umgekehrten Reihenfolge der n-1 stelligen Binärzahlen entspricht #

Satz: Sei $k \in \mathbb{N}$ mit der binären Repräsentation $(\dots b_2 b_1 b_0)_2$, dann ist $g(k) = (\dots a_2 a_1 a_0)_2$ mit

$$a_j = b_j \oplus b_{j+1}, \quad j \geq 0 \quad (*)$$

Beweis:

(IA) $n = 1$:

Fall 1) $b_0 = 0$ und somit $j = 0$.

Also $a_0 = 0 \oplus \underbrace{0}_{\text{führende Nullen!}} = 0$

$\Rightarrow$ (IA) erfüllt

Fall 2) $b_0 = 1$ und somit $j = 0$

Also $a_0 = 1 \oplus 0 = 1$

$\Rightarrow$ (IA) erfüllt

(IS) $n \to n+1$

Sei $(b_n b_{n-1} \dots b_1 b_0)_2$ eine Binärzahl mit $n+1$ Bits.

Fall $b_n = 0$: Die Graycodierung ist dann

$g(0 b_{n-1} \dots b_1 b_0) = 0 g(b_{n-1} \dots b_1 b_0)$
$= (0 a_{n-1} \dots a_1 a_0)_2$. Nach (IV) gilt
$a_j = b_j \oplus b_{j+1}$ für $j \leq n-1$.

Da $b_n = 0$ und $b_{n+1} = 0$ gilt ebenfalls $a_n = \underbrace{b_n}_{=0} \oplus \underbrace{b_{n+1}}_{=0}$,

da ja $a_n = 0$ gilt.

Fall $b_n = 1$: Also $(\underbrace{b_n \dots b_1 b_0}_{n+1 \text{ Bits}})_2 =$

$= 2^n + (\underbrace{b_{n-1} \dots b_1 b_0}_{= r < 2^n})_2$ und somit ergibt

sich $g(b_n \dots b_1 b_0) = 2^n + g(2^n - 1 - r)$

$= (a_n \dots a_1 a_0)_2$. Somit gilt $a_n = 1$.

Sei $r = (b_{n-1} \dots b_1 b_0)$, dann ist $2^n - 1 - r =$

$(\overline{b_{n-1}} \dots \overline{b_1}\, \overline{b_0})_2$, da $2^n - 1 - r$ die reflektierte

Version von r ist. Nach (IV) gilt dann

$$a_j = \overline{b_j} \oplus \overline{b_{j+1}} \overset{\text{Eigenschaft von XOR}}{=} b_j \oplus b_{j+1} \quad \text{für } j \leq n-1$$

Da $b_n = 1$ und $b_{n+1} = 0$ gilt $a_n = 1 = b_n \oplus b_{n+1}$,

d.h. auch dieser Fall ist erfüllt. #

Korollar: Sei $b = (b_n \dots b_1 b_0)_2$ eine natürliche Zahl in

Binärdarstellung, dann gilt

$$(b_n \dots b_1 b_0) \oplus \underbrace{(0\, b_n \dots b_2 b_1)}_{b/2} = g(b).$$

Nun stellt sich die Frage, wie man einen Graycode wieder in die Binärdarstellung wandelt:

<u>Satz:</u> Sei $k \in \mathbb{N}$ mit Binärdarstellung $(b_n \dots b_1 b_0)_2$ und Graycode $(a_n \dots a_1 a_0)_2$, dann gilt

$$b_j = a_j \oplus a_{j+1} \oplus \dots \oplus a_n \quad \text{für} \quad j \leq n$$

Beweis: Mit Satz (*) ergibt sich

$$b_j = \underbrace{b_j \oplus b_{j+1}}_{a_j} \oplus \underbrace{b_{j+1} \oplus \dots}_{a_{j+1}} \oplus \underbrace{b_n \oplus b_n \oplus \underbrace{b_{n+1}}_{=0}}_{a_n}$$

$$= a_j \oplus a_{j+1} \oplus \dots \oplus a_n \qquad \#$$

Eine weitere interessante Konsequenz aus Glg (*) ist folgendes Korollar:

<u>Korollar:</u> Seien $k, k' \in \mathbb{N}$, dann gilt

$$g(k \oplus k') = g(k) \oplus g(k')$$

Beweis: Direkt einsetzen und die Kommutativität von $\oplus$ ausnutzen. #

<u>Beo:</u> Aufgrund der induktiven Definition des Graycodes ist das Wort 10^* im das letzte Code wort in der Aufzählung.

Sollen nun alle Codewörter aufgezählt werden (Gray counter), dann

i) Das letzte Bit eines Graycodewortes wird getoggled, wenn alle anderen Bits 0 sind.

ii) Ein „inneres" Bit wird getoggled, wenn der Suffix hinter diesem Wort 10^* ist

iii) Das erste Bit beginnt mit 0, wird zu 1 und wird dann alle zwei Schritte geändert

<u>Bsp:</u>

		Hilfsbit
0	0 $\boxed{000}$	0
1	0001	1
2	0011	0
3	00 $\boxed{10}$	$\boxed{1}$
4	0110	0
5	0111	1
6	0101	0
7	0 $\boxed{100}$	$\boxed{1}$
8	1100	0
9	1101	1
10	1111	0
11	11 $\boxed{10}$	$\boxed{1}$
12	1010	0
13	1011	1
14	1001	0
15	1 $\boxed{000}$	$\boxed{1}$

Die folgenden Glgen beschreiben einen Graycode zähler mit n - Bit

$$Q_h^{(i+1)} = \neg Q_h^{(i)}$$ „Hilfsbit"

$$Q_0^{(i+1)} = \neg\left(Q_0^{(i)} \oplus Q_h^{(i)}\right)$$

$$Q_1^{(i+1)} = Q_1^{(i)} \oplus \Big(\underbrace{Q_0^{(i)} \wedge Q_h^{(i)}}_{\text{Togglebedingung 1 \& Hilfsbit}}\Big)$$

$$Q_2^{(i+1)} = Q_2^{(i)} \oplus \Big(\underbrace{Q_1^{(i)} \wedge \neg Q_0^{(i)} \wedge Q_h^{(i)}}_{\text{Togglebedingung 10 \& Hilfsbit}}\Big)$$

$$Q_3^{(i+1)} = Q_3^{(i)} \oplus \Big(\underbrace{Q_2^{(i)} \wedge \neg Q_1^{(i)} \wedge \neg Q_0^{(i)} \wedge Q_h^{(i)}}_{\text{Togglebedingung 100 \& Hilfsbit}}\Big)$$

$$\vdots$$

$$Q_n^{(i+1)} = Q_n^{(i)} \oplus \left(\neg Q_{k-2}^{(i)} \wedge \neg Q_{k-3}^{(i)} \wedge \ldots \wedge \neg Q_0^{(i)} \wedge Q_h^{(i)}\right)$$

Literatur:

i, Knuth, Bd 4A

ii, Wikipedia (Glgen ⇒ vgl. Alg. G von Knuth + spätere Optimierung)