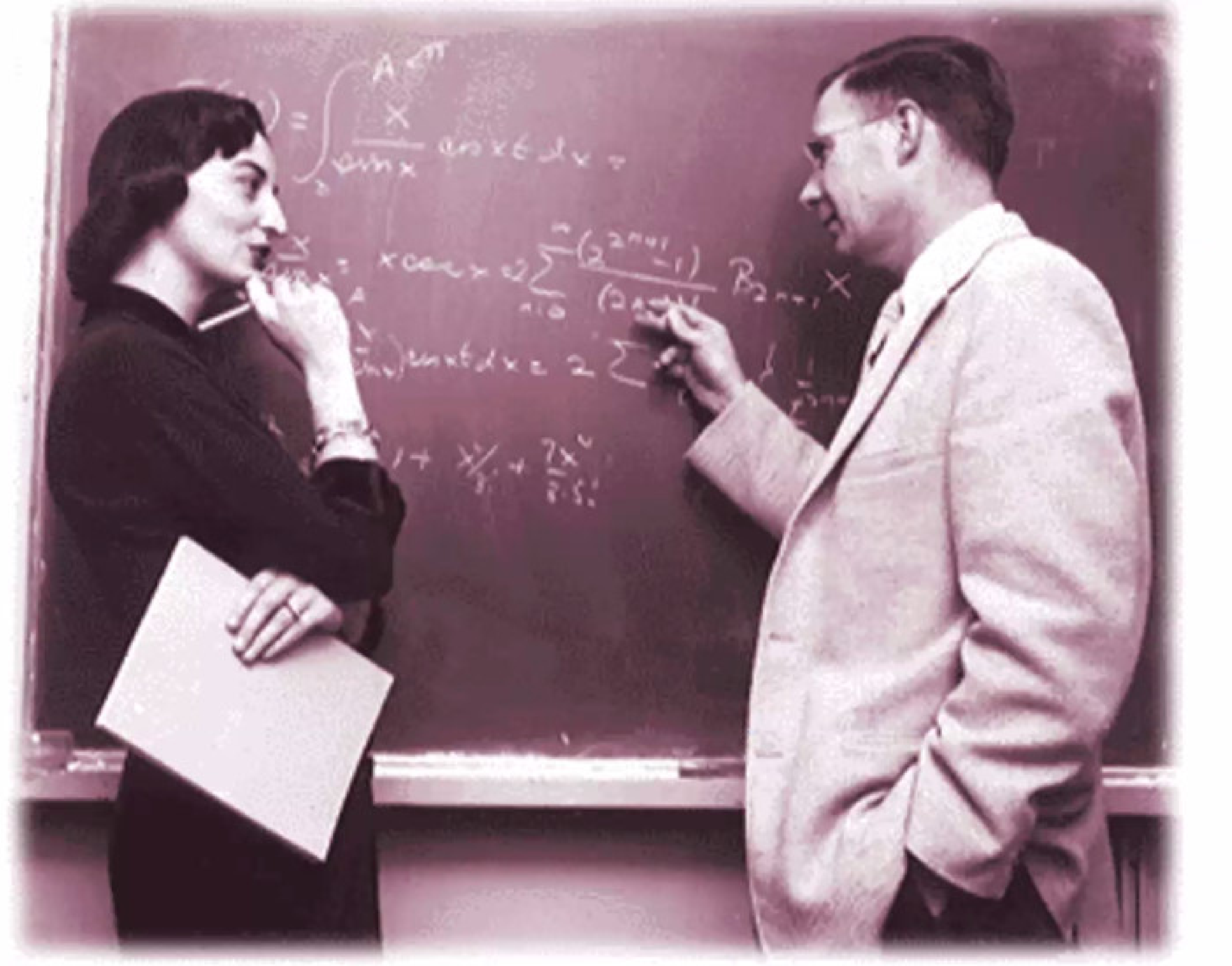
Molti altri esempi…

- Numeri di matricola delle automobili
- Codici fiscali
- Etc etc…

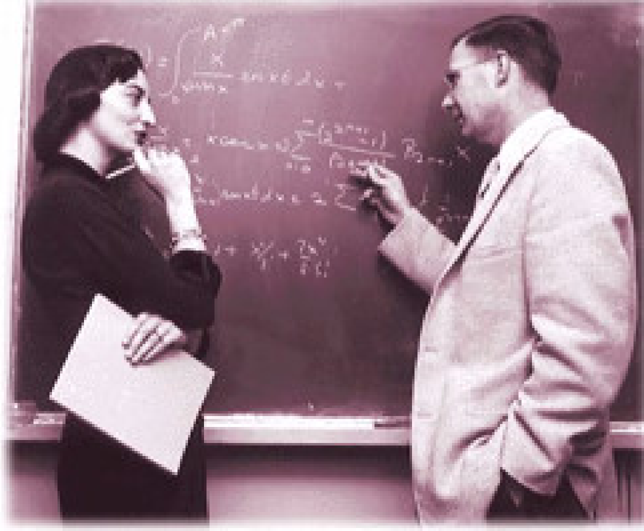
Torniamo ora ai codici di error correction

Abbiamo visto i codici di ripetizione…

- Che però sono inefficienti: ***sprecano un sacco di banda***
- Ci sono codici migliori? E in ogni caso, qual è il limite a cui possiamo aspirare?



I codici di Hamming



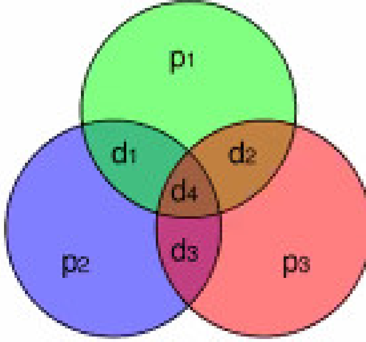
- Sono una famiglia di codici famosissima, fanno parte dei cosiddetti ***codici lineari*** perché le loro operazioni si possono esprimere tramite combinazioni lineari
- Sono sempre codici a blocco, che usano i bit di parità, solo, in modo più efficiente

Il codice Hamming (7,4)

- Il più semplice e conosciuto è il (7,4)
- Un codice Hamming **(X,Y)** significa che codifica **Y** bits di dati usandone **X** in totale
- Quindi con **X-Y** bit "extra" (in questo caso, tre bits)
- Esempio: **0100** → **0100**101

4

Notate la dilatazione dello spazio informativo:



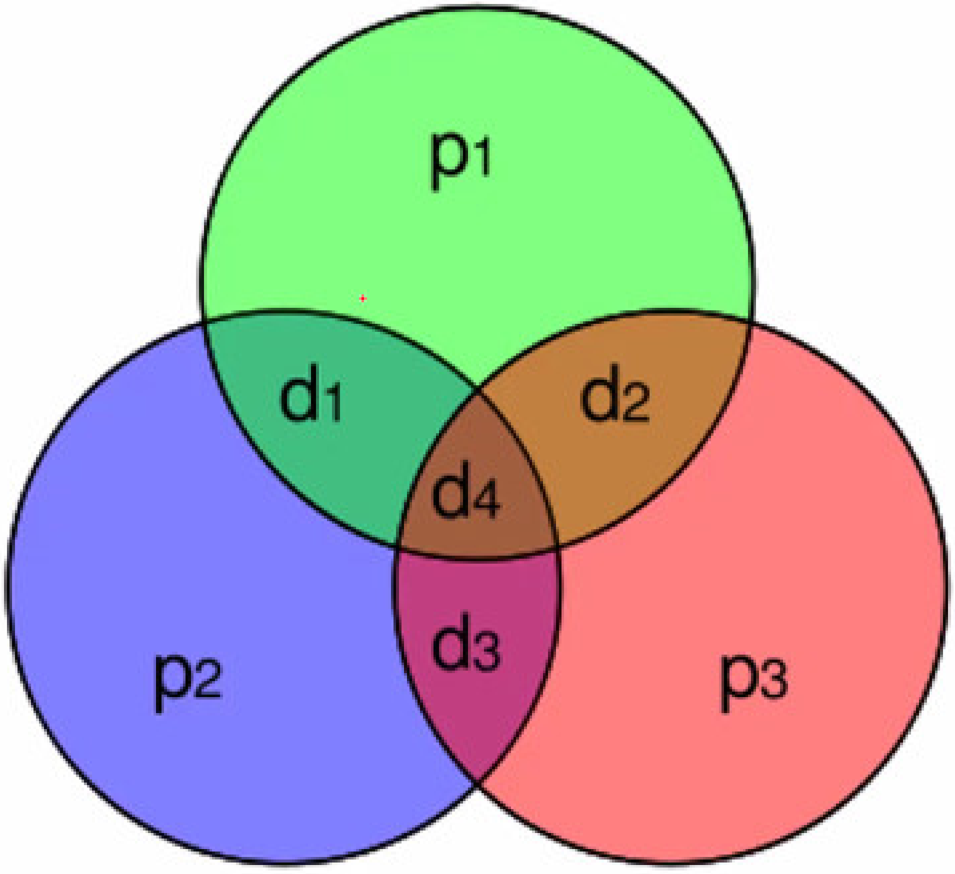
1000011
0100101
0010110
0001111
1100110
1010101
1001100
0110011
0101010
0011001
1101001
1001010
1111111
0111100
0011001
0000000

$$2^4 = 16$$ **CODICI INIZIALI**

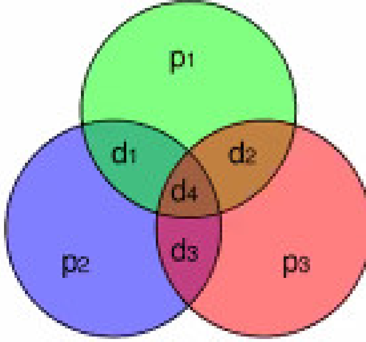
$$2^7 = 128$$ **CODICI POSSIBILI**

Qual è l'idea?

- I bit extra si "dividono" equamente la parità dei bit dati:



Esempio



- (0100)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

→ 0 1 0 0 **1 0 1**

Che proprietà hanno questi codici?

- Fanno parte della famiglia più grande, come detto, dei ***codici lineari*** (sottospazi lineari di uno spazio vettoriale su campi finiti)
- Un codice Hamming (X,Y) e che ha distanza minima di Hamming Z viene anche detto **(X,Y,Z)**

Vediamo le loro proprietà…

- Diciamo che il ***peso*** (***weight***) di un messaggio binario è il conto di quanti 1 ha
- Es. weight (1 0 0 1 1) = 3

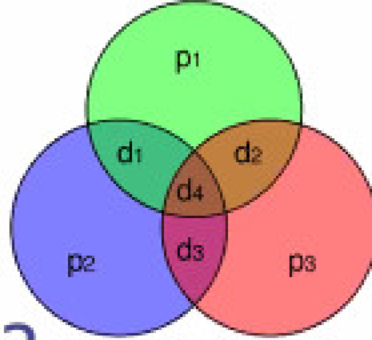
Vale allora il…

- **Teorema del peso minimo:** se il peso minimo dei vettori della matrice generatore **X** per **Y** è **d** …
- … allora ogni coppia di messaggi codificati dista almeno d
- → codice **(X,Y,d)** !

Semplicissimo!

- Quindi: prendiamo la matrice generatrice
- Calcoliamo il peso minimo **d**
- → il codice corrispondente è un codice ***error detecting*** di potenza **d-1**, e ***error correcting*** di potenza **(d-1)/2**

Esempio



- Hamming (7,4) ha potenza di correzione?

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- → PESO MINIMO = **3** → **(7,4,3)**

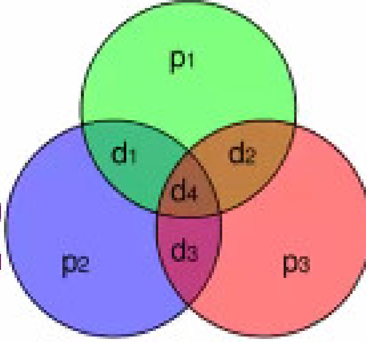
Notare…

- Quanto ***abbiamo guadagnato***: l'altro codice a potenza di correzione 1 che conoscevamo era **R_3** , che aveva come data rate **1/3 = 0.33…**
- Qui invece il data rate è **4/7 = 0.57…(!!!)**

Bella proprietà…

- … dei codici lineari, e quindi anche di quelli di Hamming:
- L'error detection si fa usando ancora operazioni lineari: una moltiplicazione per una speciale matrice, la ***matrice di parità***

Nel caso di Hamming (7,4):

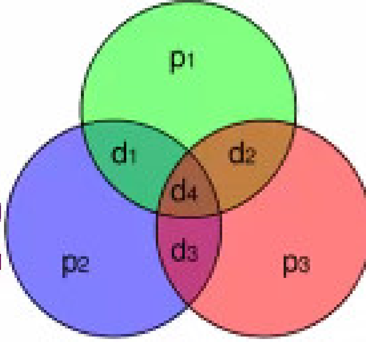


$$\mathbf{H} := \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}_{3,7}$$

E l’error recovery?

- C’è ovviamente il solito metodo del “più vicino”...
- Però pensate alla... complessità algoritmica!
- I codici lineari sono molto belli perché l’error recovery si fa in modo ***molto più veloce*** (usando la cosiddetta *sindrome*!)

Nel caso di Hamming (7,4):

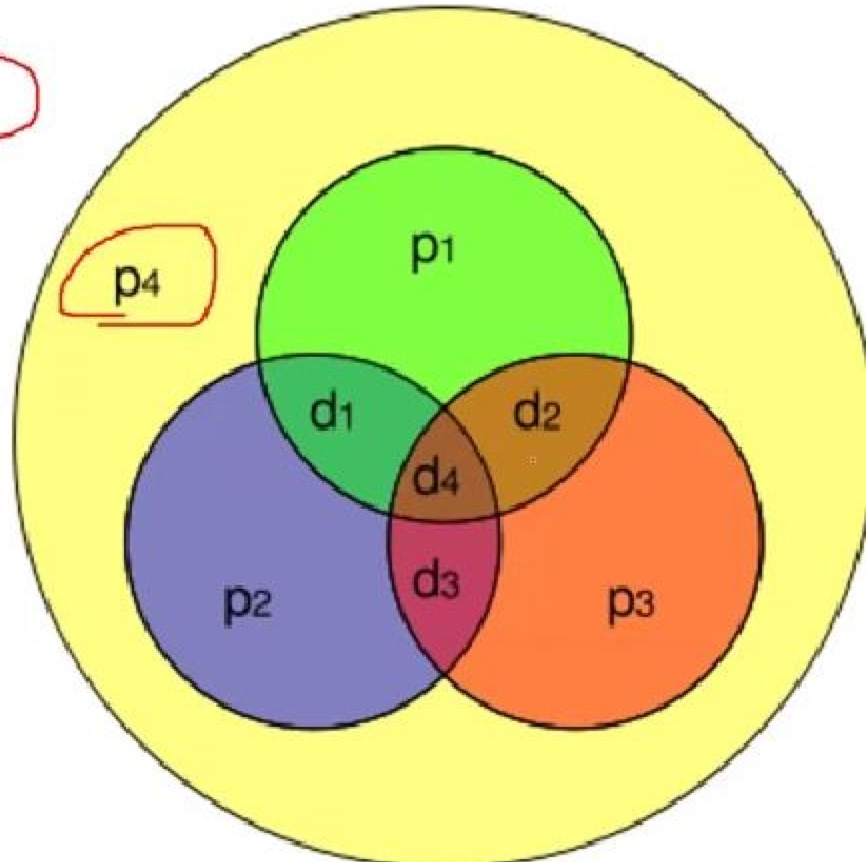


- 0 1 0 0 **1 0 1**

$$\mathbf{H} := \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}_{3,7}$$

Hamming superiori: gli Hamming "ibridi"

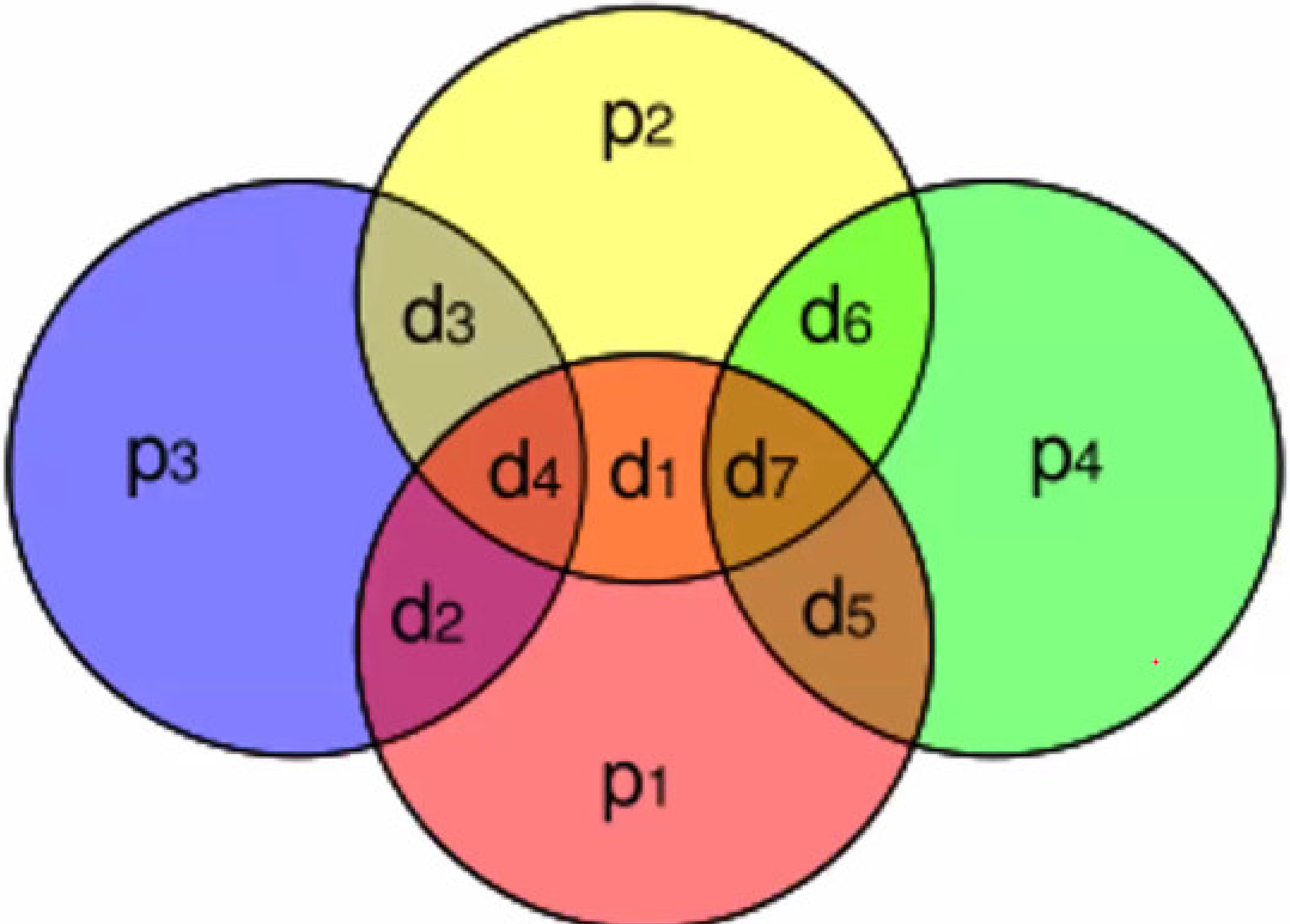
- Hamming (8,4): molto semplice, si aggiunge un altro parity bit



Proprietà di (8,4)

- La distanza minima è aumentata, stavolta a **4** (è un **(8,4,4)**)
- Quindi, ***error-detecting*** **3**, ed ***error-correcting*** **1**
- Il data rate però è sceso: da **0.57** a **0.5**

Hamming superiori: (11,7)



Proprietà di (11,7)

- Distanza minima: 5
  → codice (**(11,7,5)**)
- Quindi, **corregge fino a 2 errori (!!)**
- Data rate?
- Hamming (7,4): **0.57**
- Ora: 7/11 (circa **0.637**) !!!!

Nota… ma Hamming come ha fatto a inventarli?

Ecco il motivo… (1950!)