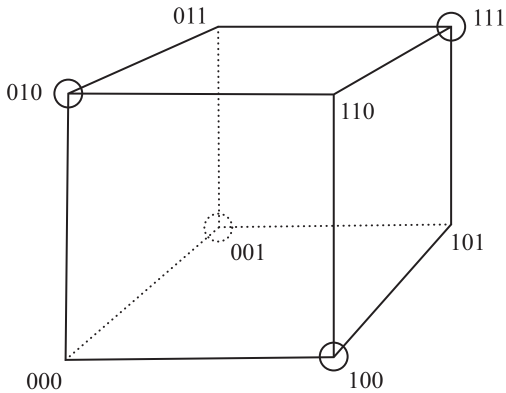

Każdy błąd w transmisji umieści nas na jednym z tych wierzchołków, mówiąc nam, że wystąpił błąd, lecz nie mówiąc gdzie. Zauważmy, że każdy fałszywy wierzchołek leży w odległości Hamminga równej 1 od prawdziwego. Gdybyśmy chcieli pojedynczej korekcji błędów dla tego systemu, musielibyśmy użyć przestrzeni dla M_C o czterech wymiarach.

Jeśli więc nasz system kodowania działa, powinniśmy być w stanie przenieść każdy z naszych punktów komunikatu gdzieś w przestrzeni komunikatów M_C, tak że będą one wystarczająco oddzielone. Od czasu do czasu będziemy zmuszeni pozwolić na pewne nakładanie się kul błędów, ale nie jest to zazwyczaj problem[5]. Możemy teraz szybko zobaczyć, jak to geometryczne podejście oferuje kolejny dowód twierdzenia Shannona. M i M_C oznaczają wymiary odpowiednio oryginalnej i zakodowanej przestrzeni komunikatów – jest to po prostu wymyślny sposób opisania długości łańcuchów w komunikacie. Liczba punktów w M wynosi 2^M, a w M_C jest to 2^{M_C}. Aby skorygować k błędów, musimy być w stanie wypełnić M_C kulami błędów o promieniu k, po jednej na każdy punkt w M. Nie chcemy, aby się one nakładały. Korzystając z tego, możemy otrzymać nierówność wiążącą objętość M_C z objętością kul. W przestrzeni dyskretnej, jaką jest przestrzeń komunikatów, objętość kuli definiuje się jako liczbę punktów w niej zawartych. Można pokazać, że dla przestrzeni M_C-wymiarowej liczba punktów oddalonych od siebie o jednostkę długości, które leżą w promieniu k jednostek od punktu, wynosi:

$$\frac{M_C!}{k!(M_C-k)!}. \tag{4.20}$$

[5] „Doskonałe” kody, które wprowadziliśmy w poprzednim problemie, są dokładnie tymi, dla których sfery błędów „wypełniają” przestrzeń komunikatów bez nachodzenia na siebie. Jeśli sfery mają promień e, to każdy punkt w przestrzeni leży wewnątrz e jednostek jednego i tylko jednego punktu komunikatu (RPF).

Zauważając, że objętość M_C musi być większa lub równa liczbie punktów w każdej kuli błędu pomnożonej przez liczbę kul, czyli liczbę punktów w M, otrzymujemy ostatecznie nierówność (4.6). Nie ma potrzeby przeprowadzania dalszych wywodów – powinniście być w stanie zobaczyć, jak pojawia się dowód.

Problem 4.3. Oto ciekawy problem, który można spróbować rozwiązać, używając przestrzeni komunikatów (tak właśnie ja to zrobiłem). Do tej pory jesteście zaznajomieni z używaniem jednego bitu parzystości na końcu komunikatu do wykrywania pojedynczych błędów. Jedną z cech tej techniki jest to, że zawsze potrzebujemy tylko jednego bitu kontrolnego, niezależnie od tego, jak długi jest komunikat – jest to niezależne od M_C. Pytanie brzmi, czy możemy również skonfigurować metodę wykrywania podwójnych błędów, która jest niezależna od M_C? Nie jesteśmy zainteresowani poprawianiem tylko wykrywaniem. Chcemy mieć skończoną liczbę bitów, zawsze taką samą, a wydaje się całkiem prawdopodobne, że możemy to zrobić, mając tylko dwa bity! Przypomnijmy, że dla kodu Hamminga mogliśmy skorygować jeden błąd i wykryć dwa błędy z bitem kontrolnym dla ogólnej parzystości i syndromu, ale liczba bitów kontrolnych wchodzących w skład syndromu zależała od długości komunikatu. Powinniście zauważyć, że faktycznie niemożliwe jest wykrycie dwóch błędów bez uwzględnienia długości komunikatu.

4.5. Kompresja danych i informacja

Za chwilę przyjrzę się twierdzeniu Shannona w jeszcze inny sposób, ale najpierw chciałbym, abyście pozwolili mi trochę zboczyć z tematu. Pierwszym kierunkiem, w którym chcę się udać, jest kompresja danych i chciałbym wyjaśnić kilka stojących za tym idei. Weźmy pod uwagę język taki jak angielski. Ma on 26 liter, a jeśli dodamy do tego przecinki, kropki, spacje i co tam jeszcze, to mamy około trzydziestu symboli do komunikacji. Więc ile rzeczy mogę powiedzieć po angielsku, jeśli mam do dyspozycji dziesięć symboli? Można by powiedzieć, no cóż, trzydzieści do potęgi dziesięć. Nie jest to prawda. Gdybym napisał następujący ciąg znaków:

cpfajarfw

to nie byłby angielski. W prawdziwym, możliwym do interpretacji angielskim, nie możemy mieć wszystkiego – liczba możliwych do przyjęcia słów jest ograniczona, a kolejność liter w nich nie jest przypadkowa. Jeśli mamy „T”, to istnieje prawdopodobieństwo, że następną literą będzie „H”. Nie będzie to „X”, a rzadko „J”. Dlaczego? Litery nie są używane równomiernie, a komunikatów w języku angielskim jest znacznie mniej, niż wydaje się na pierwszy rzut oka.