Πρόλογος

Ετούτο εδώ το βιβλίο σχεδιάστηκε για να διδαχθεί η Θεωρία Κωδίκων με έναν ορθό μαθηματικό τρόπο σε φοιτητές της Μηχανικής, της Επιστήμης των Η/Υ και των Μαθηματικών. Διαφέρει από τα πιο πολλά κείμενα με το ίδιο αντικείμενο για δυο βασικούς λόγους: ακολουθείται η φιλοσοφία του «κάθε τι στον καιρό του» και παραλείπονται οι άχρηστες μαθηματικές γενικεύσεις.

Η φιλοσοφία του «κάθε τι στον καιρό του» σημαίνει ότι η εισαγωγή των αναγκαίων μαθηματικών εννοιών γίνεται τη στιγμή που αυτές θα εφαρμοστούν· δηλ. παράλληλα με τις εφαρμογές. Δεν έχουμε 200 σελίδες μαθηματικά (τα περισσότερα να είναι άσχετα με την Θεωρία) και ακολουθούν άλλες 200 σελίδες Θεωρίας Κωδίκων. Οπότε η διάταξη της ύλης είναι περίπου: μαθηματικά, εφαρμογές, μαθηματικά, εφαρμογές, κ.ο.κ. Το να παραλείπουμε τις άχρηστες γενικεύσεις σημαίνει ότι δεν θεωρούμε αναγκαίο για παράδειγμα, να περιγράψουμε έναν κυκλικό κώδικα ως ένα πρωταρχικό ιδεώδες. Με άλλα λόγια, έχουμε παραλείψει τις περισσότερες φορές τις μαθηματικές γενικεύσεις και την ορολογία που φυσιολογικά θα χρησιμοποιούνταν στη διδασκαλία της Θεωρίας Κωδίκων σε μια τάξη η οποία αποτελείται από προχωρημένους μαθηματικούς.

Το βιβλίο μας ασχολείται αποκλειστικά με δυαδικούς κώδικες και κώδικες χαρακτηριστικής 2, δίνοντας έμφαση στην κατασκευή, κωδικοποίηση και αποκωδικοποίηση πολλών και σημαντικών οικογενειών κωδίκων που ενδιαφέρουν τη Μηχανική και την Επιστήμη των Η/Υ, όπως οι κώδικες Reed-Solomon και οι συνελικτικοί κώδικες, οι οποίοι χρησιμοποιήθηκαν στις επικοινωνίες με μακρινές αποστολές στο διάστημα αλλά και σε συνηθισμένες ηλεκτρονικές συσκευές (απλώς να αναφέρουμε δύο περιοχές εφαρμογών). Η επιλογή των κωδίκων αυτών αποτελεί την αιτία για μια μεγάλη γκάμα από αλγορίθμους για κωδικοποίηση και αποκωδικοποίηση.

Το παρόν κείμενο χρησιμοποιήθηκε για τη διδασκαλία της Θεωρίας Κωδίκων διάρκειας δυο τριμήνων στο πανεπιστήμιο του Auburn. Οι ελάχιστες προαπαιτούμενες γνώσεις για την παρακολούθηση είναι μάλλον μια στοιχειώδη γνώση Γραμμικής Άλγεβρας. Όμως όσο περισσότερες γνώσεις υπάρχουν σε Γραμμική Άλγεβρα και σε μοντέρνα Άλγεβρα γενικότερα, τόσο το καλύτερο. Φοιτητές με μεγαλύτερο μαθηματικό υπόβαθρο και ωριμότητα, θα περάσουν μάλλον γρηγορότερα την αρχική ύλη του μαθήματος.

Οι συγγραφείς θα εκτιμήσουν ιδιαίτερα τα οποιαδήποτε σχόλια από τους αναγνώστες. Η ηλεκτρονική μας διεύθυνση είναι KTPHELPS@DUCVAX.AUBURN.EDU.

Οι συγγραφείς επιθυμούν να ευχαριστήσουν την κ. Rosie Torbert για την εξαιρετική επιδεξιότητα στη δακτυλογράφηση του κειμένου. Η σταθερή και πρόσχαρη διάθεση της στις συνεχείς αναθεωρήσεις του κειμένου την ανυψώνει στην τάξη των αγίων.

D. G. Hoffman, D. A. Leonard,
C. C. Lindner, K. T. Phelps,
C. A. Rodger, J. R. Wall

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

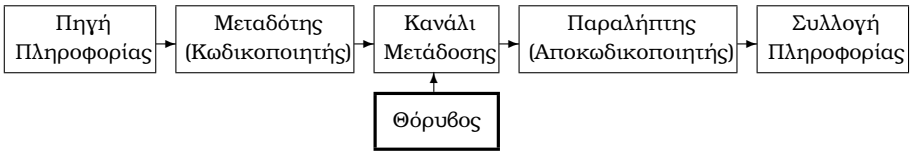
Εισαγωγή στη Θεωρία Κωδίκων

1.1 Εισαγωγή

Η Θεωρία Κωδίκων είναι η μελέτη των μεθόδων για αποτελεσματική και ακριβή μεταβίβαση πληροφοριών από το ένα μέρος στο άλλο. Η θεωρία έχει αναπτυχθεί για ποικίλες εφαρμογές, όπως ελαχιστοποίηση του θορύβου σε εγγραφές συμπαγών δίσκων, μετάδοση οικονομικών πληροφοριών μέσω τηλεφωνικών γραμμών, μετάδοση δεδομένων από τον έναν υπολογιστή στον άλλο ή από τη μνήμη στην κεντρική μονάδα επεξεργασίας και μετάδοση πληροφοριών από απομακρυσμένη πηγή, όπως μετεωρολογικούς ή επικοινωνιακούς δορυφόρους ή από το διαστημόπλοιο Voyager το οποίο στέλνει φωτογραφίες από τους πλανήτες Δία και Κρόνο στη Γη.

Το φυσικό μέσο, διαμέσου του οποίου η πληροφορία μεταδίδεται, λέγεται *κανάλι*. Οι τηλεφωνικές γραμμές και η ατμόσφαιρα είναι παραδείγματα καναλιών. Ανεπιθύμητες ενοχλήσεις, που ονομάζονται *θόρυβος*, μπορεί να προκαλέσουν διαφοροποίηση μεταξύ της μεταδιδόμενης και της ληφθείσας πληροφορίας. Ο θόρυβος μπορεί να προκληθεί από ηλιακές κηλίδες, αστραπές, τσακίσματα της μαγνητικής ταινίας, βροχές μετεωριτών, φορτωμένες τηλεφωνικές γραμμές, τυχαία ραδιοπαρεμβολή, φτωχή δακτυλογράφηση, ασαφής ακρόαση, ασαφή ομιλία και πολλές άλλες αιτίες.

Η θεωρία Κωδίκων, ασχολείται με το πρόβλημα της ανίχνευσης και της διόρθωσης λαθών από μετάδοση, τα οποία προκαλούνται από τον θόρυβο στο κανάλι. Το διάγραμμα που ακολουθεί, μας δίνει μια γενική ιδέα ενός γενικού συστήματος μετάδοσης πληροφοριών.



Το πιο σημαντικό μέρος του διαγράμματος, γι'αυτά που μας ενδιαφέρουν εδώ, είναι ο θόρυβος (Noise), ελλείψει του οποίου δε θα υπήρχε ανάγκη για τη θεωρία Κωδίκων.

Στην πράξη, ο έλεγχος που έχουμε πάνω στο θόρυβο, είναι η επιλογή ενός καλού καναλιού προς χρήση για μετάδοση και η χρήση κάποιων φίλτρων θορύβου για την αντιμετώπιση συγκεκριμένων τύπων παρεμβολής που μπορεί ν' ανακύψουν. Αυτά είναι μηχανικά προβλήματα. Έτσι και καταλήξουμε στο καλύτερο μηχανικό σύστημα προκειμένου να λύσουμε αυτά τα προβλήματα, μπορούμε να επικεντρώσουμε την προσοχή μας στην κατασκευή του κωδικοποιητή και του αποκωδικοποιητή. Η πρόθεσή μας είναι να τα κατασκευάσουμε με τέτοιο τρόπο, ώστε να πετύχουμε:

1) γρήγορη κωδικοποίηση πληροφοριών,

2) εύκολη μετάδοση των κωδικοποιημένων μηνυμάτων,

3) γρήγορη αποκωδικοποίηση των ληφθέντων μηνυμάτων,

4) διόρθωση λαθών που εισάγονται στο κανάλι και

5) μέγιστη μετάδοση πληροφοριών στη μονάδα του χρόνου.

Ο πρωταρχικός στόχος είναι ο τέταρτος από τους παραπάνω. Το πρόβλημα είναι ότι δεν είναι γενικά συμβατός με τον πέμπτο και ειδικά, ίσως δεν είναι συμβατός ούτε με τους άλλους τρεις. Έτσι, κάθε λύση επιτυγχάνεται ισορροπώντας και τους πέντε στόχους.

Στις καθημερινές προσωπικές μας επικοινωνίες, χρησιμοποιούμε βασικά λέξεις, προφορικά ή γραπτά, φτιαγμένες από περιορισμένο αλφάβητο. Έχουμε πληροφορίες για να επικοινωνούμε· τις κωδικοποιούμε σε μηνύματα που στη συνέχεια τα εκφράζουμε γραπτά ή προφορικά. Στη συνέχεια αυτά αποστέλλονται μέσω ενός καναλιού, που συνήθως είναι το διάστημα μεταξύ του στόματος και του αυτιού ή από το στυλό στο χαρτί και στη συνέχεια στο μάτι. Ο θόρυβος μπορεί να προκληθεί από ασαφή ομιλία, κακό άκουσμα, λανθασμένη γραμματική, δυνατό στερεοφωνικό συγκρότημα, φορτισμένη συζήτηση, ανορθογραφία, παρερμηνεία ή λανθασμένη δακτυλογράφηση. Ο αποκωδικοποιητής είναι το δικό μας διάβασμα (ή άκουσμα) και η κατανόηση του ληφθέντος μηνύματος.

Έχουμε εσωτερικές διαδικασίες διόρθωσης λαθών που ούτε καν είχαμε φανταστεί. Υποθέτοντας ότι λαμβάνουμε το μήνυμα "Apt natural. I have a gub.", το οποίο είναι μία ατάκα από το "Take the money and run" του Woody Allen. Αφού η Αγγλική γλώσσα δεν χρησιμοποιεί όλες τις πιθανές λέξεις οποιουδήποτε δοσμένου μήκους, πιθανότατα θα αναγνωρίσουμε ότι το "gub" δεν είναι λέξη της Αγγλικής. Μπορούμε με ασφάλεια να υποθέσουμε ότι η μεταδοθείσα λέξη είναι κοντά στο "gub" κατά μία έννοια. Άρα είναι πιο πιθανό να ήταν "gut" ή "gun" ή "tub" παρά "firetruck" ή "rat". Είναι όμως μόνο τα συμφραζόμενα του μηνύματος που μας επιτρέπουν να διαλέξουμε το "gun" σαν την πιο πιθανή λέξη. Η λέξη "Apt" είναι τέλεια ορθή στα Αγγλικά, αλλά πάλι από τα συμφραζόμενα οδηγούμαστε τελικά να τη διορθώσουμε σε "act". Αν επίσης τυχαίνει να είμαστε γνώστες της

Αγγλικής, θα διορθώσουμε επιπλέον το "natural" σε "naturally", αν και αυτό το λάθος οφείλεται στην πηγή και όχι στο κανάλι.

Από αυτούς τους τύπους λαθών, μπορούμε να ασχοληθούμε μόνο με τον πρώτο: δηλαδή με το να επιλέξουμε την πιο πιθανή μεταδοθείσα λέξη. Η σίγουρη μέθοδος για την καταπολέμηση των λαθών, είναι μέσω της χρήσης επιπλέον ψηφίων. Πολλές επιχειρήσεις σήμερα, συνήθως προσθέτουν ψηφία ελέγχου σε νούμερα αναγνώρισης. Αυτά είναι επιπλέον ψηφία που χρησιμεύουν στον έλεγχο της ορθότητας δεδομένων ή αριθμών που προκύπτουν από πράξεις. Αυτή είναι μάλλον και η πιο συνήθως αναγνωρίσιμη μέθοδος κωδικοποίησης στην καθημερινή ζωή. Θα ασχοληθούμε με πιο επιτηδευμένες παρόμοιες ιδέες.

1.2 Βασικές Υποθέσεις

Θα κάνουμε κάποιες πρωταρχικές υποθέσεις και θα δώσουμε κάποιους βασικούς ορισμούς που θα ισχύουν σε όλο το κείμενο.

Σε πολλές περιπτώσεις, η πληροφορία που πρόκειται να μεταδοθεί, μεταδίδεται από μία ακολουθία από μηδέν και άσους, τα οποία λέγονται *ψηφία*. Μία *λέξη* λοιπόν, είναι μια ακολουθία από ψηφία. *Μήκος* (length) μιας λέξης, ονομάζουμε τον αριθμό των ψηφίων που την απαρτίζουν. Έτσι, για παράδειγμα η ακολουθία 0110101 είναι μια λέξη μήκους 7. Μία λέξη μεταδίδεται στέλνοντας τα ψηφία της, το ένα μετά το άλλο, διαμέσου *δυαδικού καναλιού (binary channel)*. Ο όρος «δυαδικό» αναφέρεται στο γεγονός ότι μόνο δύο ψηφία, το μηδέν και ο άσος, χρησιμοποιούνται. Κάθε ψηφίο μεταδίδεται με τρόπο μηχανικό, ηλεκτρικό, μαγνητικό, ή αλλιώς, με έναν από τους δύο τύπους εύκολα διαφοροποιούμενων παλμών.

Ένας *δυαδικός κώδικας (binary channel)* είναι ένα σύνολο C από λέξεις. Ο κώδικας που αποτελείται απ' όλες τις λέξεις μήκους 2, είναι ο

$$C = \{00, 10, 01, 11\}.$$

Ένας *μπλοκ κώδικας* (block code) είναι ένας κώδικας του οποίου όλες οι λέξεις έχουν το ίδιο *μήκος*· αυτό το μήκος ονομάζεται *μήκος* του κώδικα. Εμείς θα θεωρούμε στο εξής μόνο μπλοκ κώδικες. Δηλαδή, για μας, ο όρος *κώδικας* θα σημαίνει πάντα δυαδικός μπλοκ κώδικας. Οι λέξεις που ανήκουν σε δοσμένο κώδικα C_0, θα λέγονται *κωδικολέξεις*. Θα συμβολίζουμε τον αριθμό των κωδικολέξεων σε έναν κώδικα C, με $|C|$.

Ασκήσεις

1.2.1 Κάντε μια λίστα με όλες τις λέξεις μήκους 3, μήκους 4, και μήκους 5.

1.2.2 Βρείτε έναν τύπο για το συνολικό αριθμό των λέξεων μήκους n.

1.2.3 Έστω C είναι ο κώδικας που αποτελείται από όλες τις λέξεις μήκους 6 που έχουν άρτιο πλήθος από άσους. Κάντε μια λίστα με τις κωδικολέξεις του C.

Θα χρειαστούμε να κάνουμε κάποιες συγκεκριμένες βασικές υποθέσεις για το κανάλι. Αυτές οι υποθέσεις θα μορφοποιήσουν αναγκαστικά την θεωρία που σχηματίζουμε.

Η πρώτη υπόθεση είναι ότι μια κωδικολέξη μήκους n που αποτελείται από 0 και 1 παραλαμβάνεται ως μια λέξη μήκους n που αποτελείται από 0 και 1, αλλά δεν αποκλείεται να είναι διαφορετική από τη σταλθείσα.

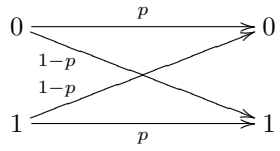
Η δεύτερη είναι ότι δεν υπάρχει δυσκολία να αναγνωρίσουμε την αρχή της πρώτης σταλθείσας λέξης. Έτσι, εάν χρησιμοποιούμε κωδικολέξεις μήκους 3 και λάβουμε 011011001, τότε ξέρουμε ότι λάβαμε τις παρακάτω 3 λέξεις κατά σειρά: 011, 011, 001. Αυτή η υπόθεση σημαίνει, εάν χρησιμοποιούμε μήκος 3, ότι το κανάλι δεν μπορεί να παραδώσει 01101 στον αποκωδικοποιητή, διότι ένα ψηφίο έχει χαθεί εδώ.

Η τελική υπόθεση είναι ότι ο θόρυβος απλώνεται ισοπίθανα και ομοιογενώς σε αντίθεση με το να απλώνεται σε δέσμες (ριπές) που λέγονται *εκρήξεις*. Δηλαδή η πιθανότητα ένα τυχαίο ψηφίο να είναι εσφαλμένο κατά τη λήψη του (λόγο θορύβου) είναι η ίδια ακριβώς με οποιοδήποτε έλλο ψηφίο και δεν επηρεάζεται από λάθη σε γειτονικά ψηφία. Αυτή δεν είναι μια πολύ ρεαλιστική υπόθεση για πολλά είδη θορύβου όπως οι αστραπές και οι γρατζουνιές σε συμπαγείς δίσκους. Θα θεωρήσουμε τελικά αυτό το είδος θορύβου.

Σε ένα *τέλειο*, δηλαδή χωρίς θορύβους, κανάλι, κάθε ψηφίο 0 ή 1 που στέλνεται, είναι πάντοτε αυτό που λαμβάνεται. Εάν όλα τα κανάλια ήταν τέλεια, δε θα είχαμε ανάγκη τη Θεωρία Κωδίκων. Όμως ευτυχώς (ή ίσως, δυστυχώς) κανένα κανάλι δεν είναι τέλειο· κάθε κανάλι είναι θορυβώδες. Κάποια κανάλια είναι λιγότερα θορυβώδη, ή πιο αξιόπιστα, από άλλα.

Ένα δυαδικό κανάλι θα λέγεται *συμμετρικό*, αν τα 0 και 1 λαμβάνονται με την ίδια πιθανότητα λάθους· δηλαδή η πιθανότητα να λαμβάνουμε το σωστό ψηφίο είναι ανεξάρτητη από το ποιο ψηφίο, 0 ή 1, αποστέλλεται. Η *αξιοπιστία* ενός δυαδικού συμμετρικού καναλιού (ΔΣΚ)(binary symmetric channel) είναι ένας πραγματικός αριθμός p, με $0 \leq p \leq 1$, ο οποίος συμβολίζει την πιθανότητα το ψηφίο $(0$ ή $1)$ που λαμβάνεται να είναι εκείνο ακριβώς που έχει αποσταλεί.

Εάν το p συμβολίζει την πιθανότητα το ψηφίο που λαμβάνεται να είναι το ίδιο με το ψηφίο που μας στάλθηκε, τότε $1-p$ είναι η πιθανότητα το ψηφίο που λαμβάνεται να μην είναι εκείνο που στάλθηκε. Το παρακάτω διάγραμμα αποσαφηνίζει πώς λειτουργεί ένας ΔΣΚ:



Στις περισσότερες περιπτώσεις είναι ίσως δύσκολο να υπολογίσουμε την ακριβή τιμή του p για δοθέν κανάλι. Εντούτοις η πραγματική τιμή του p δεν επηρεάζει σημαντικά την ανάπτυξη της θεωρίας.

Ένα κανάλι ονομάζεται πιο αξιόπιστο από ένα άλλο, αν η αξιοπιστία του είναι μεγαλύτερη. Παρατηρούμε ότι για $p = 1$, δεν υπάρχει περίπτωση κάπιο ψηφίο να αλλαχθεί κατά την μετάδοση. Οπότε το κανάλι είναι τέλειο και δε θα μας απασχολήσει. Όπως επίσης δεν θα μας απασχολήσει και ένα κανάλι με $p = 0$.

Κάθε κανάλι με $0 < p \leq 1/2$, μπορεί εύκολα να μετατραπεί σε ένα άλλο κανάλι με $1/2 \leq p < 1$. *Από 'δώ και στο εξής θα υποθέτουμε πάντα ότι χρησιμοποιούμε ένα ΔΣΚ με αξιοπιστία p που ικανοποιεί τη σχέση $1/2 < p < 1$.*

Ασκήσεις

1.2.4 Εξηγήστε γιατί ένα κανάλι με $p = 0$ δεν έχει ενδιαφέρον.

1.2.5 Εξηγήστε πώς μετατρέπεται ένα κανάλι με πιθανότητα $0 < p \leq 1/2$, σε κανάλι με πιθανότητα $1/2 \leq p < 1$.

1.2.6 Τι μπορούμε να πούμε για ένα κανάλι με $p = 1/2$;

1.3 Ανιχνεύοντας και διορθώνοντας υποδείγματα λαθών

Θεωρούμε τώρα τις δυνατότητες διόρθωσης και ανίχνευσης λαθών.

Σ' αυτήν την ενότητα θα αναπτύξουμε διαισθητικά τις έννοιες που συνεπάγεται η ανίχνευση και η διόρθωση λαθών, ενώ μια πλήρης ανάπτυξη θα ακολουθήσει στις επόμενες παραγράφους.

Ας υποθέσουμε λοιπόν ότι μια λέξη παραλήφθηκε, η οποία όμως δεν είναι κωδικολέξη. Συνεπώς *ανιχνεύσαμε* ότι κάποιο λάθος (ίσως αρκετά λάθη) έχει εισαχθεί. Αν αντίθετα λάβουμε μια κωδικολέξη, τότε είναι πιθανό να μην υπάρχουν λάθη κατά τη διάρκεια της μετάδοσης, οπότε είναι αδύνατο να ανιχνεύσουμε κάποιο λάθος.

Η έννοια της διόρθωσης ενός λάθους είναι πιο περίπλοκη. Όπως είδαμε και στο παράδειγμα στην εισαγωγή όταν κλίναμε να διορθώσουμε το "gub" σε "gun" παρά σε "rat", απευθυνθήκαμε στη διαίσθηση μας για να προτείνουμε ότι κάθε ληφθείσα λέξη θα πρέπει να *διορθωθεί* σε μια κωδικολέξη που απαιτεί τις λιγότερες αλλαγές. (Σε επόμενη ενότητα δείχνουμε ότι η πιθανότητα ώστε μια τέτοια κωδικολέξη αποστάλθηκε είναι τουλάχιστον τόσο μεγάλη όσο η πιθανότητα να μας αποστάλθηκε οποιαδήποτε άλλη λέξη). Για να ενώσουμε αυτές τις ιδέες, θα μελετήσουμε κάποιους συγκεκριμμένους κώδικες. Σημειώστε ότι η υπόθεσή μας ότι αποκλείουμε την περίπτωση να χάνονται ή να δημιουργούνται ψηφία κατά τη διάρκεια της μετάδοσης αποκλείει την αποκωδικοποίηση του "gub" σε "firetruck".

Παράδειγμα 1.3.1 Έστω ο κώδικας $C_1 = \{00, 01, 10, 11\}$. Τότε κάθε παραληφθείσα λέξη είναι κωδικολέξη και ο C_1 δεν μπορεί ν' ανιχνεύσει κάποιο λάθος. Επίσης ο C_1 δε διορθώνει κάποια λάθη διότι κάθε ληφθείσα λέξη δεν απαιτεί καμία αλλαγή για να γίνει κωδικολέξη.

Παράδειγμα 1.3.2 Τροποποιούμε τον C_1 επαναλαμβάνοντας κάθε κωδικολέξη τρεις φορές. Ο νέος κώδικας είναι ο

$$C_2 = \{000000, 010101, 101010, 111111\}.$$

Αυτό είναι ένα παράδειγμα ενός *επαναληπτικού* κώδικα. Έστω τώρα ότι λαμβάνουμε 110101. Επειδή δεν είναι κωδικολέξη μπορούμε να ανιχνεύσουμε ότι τουλάχιστον ένα λάθος εμφανίστηκε. Η κωδικολέξη 010101 μπορεί να σχηματιστεί αλλάζοντας ένα ψηφίο, αλλά όλες οι άλλες κωδικολέξεις σχηματίζονται αλλάζοντας περισσότερα από ένα ψηφία. Οπότε περιμένουμε ότι η 010101 είναι η πιο πιθανή κωδικολέξη που μας έχει αποσταλεί, οπότε διορθώνουμε την 110101 σε 010101. (Μια κωδικολέξη που μπορεί να σχηματιστεί από μια λέξη w με το ελάχιστο πλήθος ψηφίων να αλλάζεται λέγεται η *κοντινότερη* κωδικολέξη· η ιδέα θα σχηματοποιηθεί αργότερα). Πράγματι εάν μια οποιαδήποτε κωδικολέξη $c \in C_2$, μεταδίδεται και ένα λάθος εμφανίζεται κατά την διάρκεια της μετάδοσης, τότε η μοναδική κοντινότερη κωδικολέξη στην παραληφθείσα λέξη είναι η c· οπότε κάθε ένα λάθος επιδρά σε μια λέξη την οποία διορθώνουμε στην κωδικολέξη που μεταδόθηκε.

Παράδειγμα 1.3.3 Τροποποιούμε τον C_1 προσθέτοντας ένα τρίτο ψηφίο σε κάθε κωδικολέξη έτσι ώστε το πλήθος των άσων σε κάθε κωδικολέξη να είναι άρτιο. Ο κώδικας που προκύπτει είναι ο

$$C_3 = \{000, 011, 101, 110\}.$$

Το επιπλέον ψηφίο λέγεται ψηφίο *ελέγχου της ισοτιμίας* (*parity-check* ψηφίο). Έστω ότι λάβαμε 010, τότε επειδή η 010 δεν είναι κωδικολέξη, έχουμε ανιχνεύσει ότι κάποιο λάθος έχει συμβεί. Κάθε μια από τις κωδικολέξεις $110, 000$ και 011 μπορεί να σχηματιστεί με την αλλαγή ενός ψηφίου από τη ληφθείσα λέξη. Σε επόμενες ενότητες διακρίνουμε μεταξύ της περίπτωσης του πως χειριζόμαστε ληφθείσες λέξεις που είναι κοντινότερα σε μια μοναδική κωδικολέξη (και άρα είναι η μοναδικά πιο πιθανή κωδικολέξη που αποστάλθηκε) όπως ήταν στο Παράδειγμα 1.3.2 και της περίπτωσης που οι ληφθείσες λέξεις είναι κοντινότερα σε πολλές κωδικολέξεις όπως σε τούτο το παράδειγμα. Είναι αρκετό σε αυτό το στάδιο να παρατηρήσουμε ότι φαίνεται πιο λογικό να διορθώσουμε την 010 σε μια από τις $110, 000$ ή 011 παρά σε 101.

Ασκήσεις

1.3.4 Έστω C ο κώδικας που περιέχει όλες τις κωδικολέξεις μήκους 3. Βρείτε ποια κωδικολέξη είναι πιο πιθανό να έχει σταλεί, αν λάβουμε την 001.

1.3.5 Προσθέστε ένα parity check ψηφίο στις κωδικολέξεις του κώδικα της άσκησης 1.3.4 και χρησιμοποιείστε τον κώδικα C που προκύπτει για να απαντήσετε στις παρακάτω ερωτήσεις.

(α΄). Εάν λάβουμε την 1101, μπορούμε ν' ανιχνεύσουμε ένα λάθος;

(β΄). Εάν λάβουμε την 1101, τι κωδικολέξεις είναι πιο πιθανό να έχουν μεταδοθεί;

(γ΄). Υπάρχει κάποια λέξη με μήκος 4 η οποία να μην ανήκει στον κώδικα, αλλά να είναι κοντά σε μια μοναδική κωδικολέξη;

1.3.6 Επαναλάβετε κάθε λέξη στον κώδικα C που περιγράφεται στην άσκηση 1.3.4 τρεις φορές, διατυπώνοντας έναν επαναληπτικό κώδικα μήκους 9. Βρείτε τις κοντινότερες κωδικολέξεις στις ακόλουθες ληφθείσες λέξεις:

(α'). 001000001

(β'). 011001011

(γ'). 101000101

(δ'). 100000010

1.3.7 Βρείτε το μέγιστο αριθμό κωδικολέξεων μήκους $n = 4$ σε έναν κώδικα, στον οποίο κάθε μοναδικό λάθος μπορεί να ανιχνευθεί.

1.3.8 Επαναλάβετε την άσκηση 1.3.7 για $n = 5$, $n = 6$ και για κάθε n.

1.4 Βαθμός Πληροφορίας

Μετά την τελευταία ενότητα έγινε φανερό ότι η πρόσθεση νέων ψηφίων σε κωδικολέξεις, βελτιώνουν τις δυνατότητες ανίχνευσης και διόρθωσης λαθών του κώδικα. Ωστόσο, όσο μακρύτερη είναι η κωδικολέξη, τόσο πιο αργή και η αποστολή της. Ο *βαθμός πληροφορίας* (ή απλά βαθμός), ενός κώδικα είναι ένας αριθμός που σχεδιάστηκε να μετράει το ποσοστό από κάθε κωδικολέξη που μεταφέρει το μήνυμα. Ο βαθμός πληροφορίας για έναν κώδικα C με μήκους n, ορίζεται να είναι ο λόγος (για δυαδικούς κώδικες)

$$\frac{1}{n}\log_2 |C|.$$

Επειδή μπορούμε να υποθέσουμε ότι $1 \leq |C| \leq 2^n$, είναι φανερό ότι ο βαθμός πληροφορίας είναι μεταξύ 0 και 1· είναι 1 εάν κάθε λέξη είναι και κωδικολέξη και 0 αν $|C| = 1$.

Για παράδειγμα, οι βαθμοί πληροφορίας για τους κώδικες C_1, C_2 και C_3 της προηγούμενης ενότητας, είναι 1, $1/3$ και $2/3$ αντίστοιχα. Καθένας απ' αυτούς τους βαθμούς πληροφορίας φαίνεται λογικό να συσχετίζεται με τους αντίστοιχους κωδίκους, διότι μόνο τα πρώτα 2 από τα 6 ψηφία κάθε μιας κωδικολέξης του C_2 μεταφέρουν το μήνυμα, όπως και τα πρώτα 2 ψηφία από τα 3 κάθε κωδικολέξης του C_3.

Ασκήσεις

1.4.1 Βρείτε το βαθμό πληροφορίας για κάθε κώδικα στις Ασκήσεις 1.3.4, 1.3.5 και 1.3.6.

1.5 Τα θετικά αποτελέσματα από την ανίχνευση και τη διόρθωση λαθών

Για να τονίσουμε τα πολύ θετικά αποτελέσματα από την εισαγωγή του parity-check ψηφίου σ' ένα κώδικα στην ανίχνευση όταν εμφανιστεί ένα λάθος, θεωρούμε τους παρακάτω κώδικες.

Έστω ότι όλες οι 2^{11} λέξεις μήκους 11 είναι κωδικολέξεις, οπότε κανένα λάθος δεν πρόκειται να ανιχνευθεί. Έστω ότι η αξιοπιστία του καναλιού είναι $p = 1-10^{-8}$ και ότι τα ψηφία αποστέλλονται με ρυθμό 10^7 ψηφία το δευτερόλεπτο. Τότε, η πιθανότητα μια λέξη να μεταδίδεται εσφαλμένη είναι ίση περίπου με $11p^{10}(1-p)$, που είναι περίπου $11/10^8$. Οπότε περίπου

$$\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1 \text{ λέξεις το δευτερόλεπτο}$$

αποστέλλονται εσφαλμένα, χωρίς να ανιχνεύονται. Δηλαδή μια ολόκληρη λέξη κάθε 10 δευτερόλεπτα, δηλαδή 6 λέξεις το λεπτό, 360 λέξεις την ώρα ή 8640 τη μέρα! Αυτό δεν είναι πολύ καλό.

Ας υποθέσουμε τώρα ότι εισάγουμε ένα parity-check ψηφίο σε κάθε κωδικολέξη, οπότε το πλήθος των άσων σε κάθε μία από τις 2048 κωδικολέξεις είναι άρτιο. Άρα κάθε λέξη με ένα λάθος ανιχνεύεται, οπότε τουλάχιστον δύο λάθη πρέπει να εμφανίζονται σε κάποια λέξη, ώστε αυτή να μεταδίδεται λανθασμένα χωρίς να το αντιληφθούμε. Η πιθανότητα τουλάχιστον δύο λάθη να εμφανίζονται είναι ίση με $1 - p^{12} - 12p^{11}(1-p)$, το οποίο είναι περίπου $\binom{12}{2} p^{10}(1-p)^2$, το οποίο για $p = 1 - 10^{-8}$ είναι περίπου $\frac{66}{10^{16}}$. Τώρα περίπου $\frac{66}{10^{16}} \cdot \frac{10^7}{12} = 5 \cdot 5 \times 10^{-9}$ λέξεις το δευτερόλεπτο μεταδίδονται λανθασμένες, χωρίς ν' ανιχνεύονται. Αυτό σημαίνει μία λανθασμένη λέξη κάθε 2000 μέρες!

Έτσι λοιπόν, εάν επιθυμούμε να ελαττώσουμε λίγο το βαθμό πληροφορίας, επιμηκύνοντας το κώδικα από 11 σε 12, τότε είναι πολύ πιθανό ν' ανακαλύπτουμε τα λάθη. Για να αποφασίσουμε αν όντως αυτά τα λάθη συνέβησαν, ίσως χρειαστεί να ζητήσουμε την επαναμετάδοση του μηνύματος. Αυτό σημαίνει, από πρακτικής απόψεως, ότι η μετάδοση πρέπει να σταματήσει έως ότου λάβουμε την επιβεβαίωση, ή τα μηνύματα πρέπει να αποθηκεύονται κάπου προσωρινά, έως ότου απαιτήσουμε επαναμετάδοση· και οι δύο περιπτώσεις μπορεί να έχουν υψηλό κόστος σε χρόνο ή χώρο. Υπάρχουν βέβαια και οι περιπτώσεις που η επανάληψη είναι αδύνατη, όπως για παράδειγμα στην αποστολή του Voyager και όταν χρησιμοποιούμε συμπαγείς δίσκους. Οπότε, αντί να μεγαλώσουμε το μήκος μιας λέξης, αξίζει να εισάγουμε ικανότητες διόρθωσης λαθών μέσα στον ίδιο τον κώδικα. Τέτοιες όμως ικανότητες μπορεί να δυσκολέψουν την κωδικοποίηση, αλλά θα βοηθήσουν να αποφύγουμε το υψηλό κόστος σε χρόνο ή χρήμα που προαναφέραμε.

Ένας άλλος τρόπος για να εισάγουμε ικανότητες διόρθωσης, είναι να φτιάξουμε έναν επαναληπτικό κώδικα στον οποίο κάθε κωδικολέξη αποστέλλεται τρεις φορές διαδοχικά. Οπότε, αν υποθέσουμε ότι το πολύ ένα λανθασμένο ψηφίο εμφανίζεται σε κάθε κωδικολέξη των 33 ψηφίων, τότε δύο τουλάχιστον από τις τρεις επαναλήψεις θα είναι σωστές. Επειδή οι συγκρίσεις των τριών λέξεων των 11 ψη-

φίων είναι σχετικά απλό, το μόνο μειονέκτημα είναι ότι ο βαθμός πληροφορίας μειώθηκε από 1 σε $1/3$.

Όμως το $1/3$ παραμένει $1/3$. Ίσως υπάρχει και καλύτερος τρόπος. Παρακάτω θα δούμε ότι είναι δυνατό, εισάγοντας 4 μόνο επιπλέον ψηφία σε κάθε κωδικολέξη των 11 ψηφίων, να διορθώσουμε κάθε μοναδικό λανθασμένο ψηφίο. Αυτό σημαίνει ότι έχουμε ένα κώδικα με βαθμό πληροφορίας $11/15$, μια αξιοσημείωτη βελτίωση, υποθέτοντας βέβαια ότι το κόστος κωδικοποίησης και αποκωδικοποίησης δεν αυξάνει υπερβολικά.

Το καθήκον μας λοιπόν είναι να σχεδιάσουμε κώδικες με λογικούς βαθμούς πληροφορίας, χαμηλό κόστος κωδικοποίησης και αποκωδικοποίησης και με κάποιες ικανότητες ανίχνευσης και διόρθωσης λαθών, ώστε να μην απαιτείται επανάληψη της μετάδοσης του μηνύματος.

1.6 Βρίσκοντας την πιο πιθανή κωδικολέξη που μεταδόθηκε

Ας υποθέσουμε ότι έχουμε μια συνολική αντίληψη της διαδικασίας μετάδοσης, γνωρίζοντας και την κωδικολέξη v που αποστάλθηκε και την κωδικολέξη w που παραλήφθηκε. Για δοθέντα v και w, έστω $\phi_p(v, w)$ η πιθανότητα εκείνη που, αν η κωδικολέξη v αποστάλθηκε μέσω ενός ΔΣΚ με αξιοπιστία p, τότε η λέξη w παραλήφθηκε. Επειδή υποθέτουμε ότι ο θόρυβος απλώνεται τυχαία, μπορούμε να μεταχειριζόμαστε τη μετάδοση κάθε ψηφίου ως ανεξάρτητο γεγονός. Οπότε, αν οι v και w διαφωνούν σε d ψηφία, τότε έχουμε $n - d$ ψηφία να έχουν μεταδοθεί σωστά και d να έχουν μεταδοθεί λανθασμένα, οπότε,

$$\phi_p(v, w) = p^{n-d}(1 - p)^d.$$

Παράδειγμα 1.6.1 Έστω C κώδικας μήκους 5. Τότε για κάθε v στο C, η πιθανότητα να έχει αποσταλεί σωστά είναι:

$$\phi_p(v, v) = p^5.$$

Έστω 10101 ανήκει στο C. Άρα:

$$\phi_p(10101, 01101) = p^3(1 - p)^2$$

και αν $p = 0.9$ τότε:

$$\phi_{0.9}(10101, 01101) = (0.9)^3(0.1)^2 = 0.00729.$$

Ασκήσεις

1.6.2 Υπολογίστε την $phi_{0.97}(v, w)$ για κάθε ένα απο τα παρακάτω ζεύγη των v και w:

(α'). $v = 01101101$, $w = 10001110$

(β΄). $v = 1110101$, $w = 1110101$

(γ΄). $v = 00101$, $w = 11010$

(δ΄). $v = 00000$, $w = 00000$

(ε΄). $v = 1011010$, $w = 0000010$

(ϛ΄). $v = 10110$, $w = 01001$

(ζ΄). $v = 111101$, $w = 000010$.

Στην πράξη, γνωρίζουμε την w, την λέξη που λάβαμε αλλά δεν γνωρίζουμε την πραγματικά απεσταλμένη κωδικολέξη v. Εντούτοις, κάθε κωδικολέξη v ορίζει για τις λέξεις w μια αντιστοιχία από πιθανότητες $\phi_p(v, w)$. Κάθε τέτοια αντιστοιχία είναι ένα μαθηματικό μοντέλο και επιλέγουμε το μοντέλο (δηλαδή την κωδικολέξη v) το οποίο συμφωνεί περισσότερο με την παρατήρηση - σ' αυτήν την περίπτωση, εκείνη που κάνει πιο πιθανή την παραληφθείσα λέξη. Δηλαδή, υποθέτουμε ότι η v αποστάλθηκε εάν η w παραλήφθηκε, όταν

$$\phi_p(v, w) = \max\{\phi_p(u, w) : u \in C\}.$$

Το παρακάτω θεώρημα μας δίνει ένα απλό κριτήριο για να βρίσκουμε τέτοια κωδικολέξη v.

Θεώρημα 1.6.3 *Έστω ότι έχουμε ένα ΔΣΚ με $1/2 < p < 1$. Έστω v_1 και v_2 δύο κωδικολέξεις και w μια λέξη, όλες μήκους n. Υποθέτουμε ότι οι v_1 και w διαφωνούν σε d_1 ψηφία και οι v_2 και w σε d_2. Τότε:*

$$\phi_p(v_1, w) \leq \phi_p(v_2, w) \text{ αν και μόνο αν } d_1 \geq d_2.$$

Απόδειξη: Έχουμε ήδη αποδείξει ότι

$$\begin{aligned} \phi_p(v_1, w) \leq \phi_p(v_2, w) &\Leftrightarrow p^{n-d_1}(1-p)^{d_1} \leq p^{n-d_2}(1-p)^{d_2} \\ &\Leftrightarrow \left(\frac{p}{1-p}\right)^{d_2-d_1} \leq 1 \\ &\Leftrightarrow d_2 \leq d_1 \text{ (εφόσον } \frac{p}{1-p} > 1). \end{aligned}$$

Το προηγούμενο μας δίνει μία απλή μέθοδο για να διορθώσουμε λέξεις, την οποία μέχρι τώρα την είχαμε αποδεχθεί διαισθητικά: διόρθωσε τη w σε μια κωδικολέξη που διαφωνεί με τη w σε όσο το δυνατόν λιγότερα ψηφία, διότι μία τέτοια κωδικολέξη είναι η πιο πιθανή να έχει σταλεί, δεδομένου ότι παραλήφθηκε η w.

Παράδειγμα 1.6.4 Εάν $w = 00110$ έχει παραληφθεί μέσω ενός ΔΣΚ με $p = 0.98$, ποια από τις παρακάτω κωδικολέξεις 01101, 01001, 10100, 10101 είναι η

πιθανότερη αποσταλθείσα;

v	d (πλήθος ψηφίων που διαφωνούν με τη w)
01101	3
01001	4
10100	2 $\leftarrow$ το μικρότερο d
10101	3

Χρησιμοποιώντας τον παραπάνω πίνακα, το θεώρημα 1.6.3 μας λέει ότι η 10100 ήταν η πιο πιθανή σταλθείσα λέξη. Σημειώστε ότι δεν είναι ανάγκη να γνωρίζουμε την ακριβή τιμή του p για να εφαρμόσουμε το θεώρημα 1.6.3· το μόνο που πρέπει να γνωρίζουμε είναι ότι $p > 1/2$.

Ασκήσεις

1.6.5 Υποθέστε ότι η $w = 0010110$ παραλήφθηκε από ένα ΔΣΚ κανάλι με αξιοπιστία $p = 0.9$. Ποια από τις παρακάτω κωδικολέξεις είναι πιο πιθανό να έχει αποσταλεί;

$$1001011, 111110, 0001110, 0011001, 1101001.$$

1.6.6 Ποια από τις 8 κωδικολέξεις του κώδικα στην άσκηση 1.3.6 είναι η πιο πιθανή να έχει αποσταλεί, αν έχουμε λάβει $w = 101000101$;

1.6.7 Εάν $C = \{01000, 01001, 00011, 11001\}$ και λάβαμε μια λέξη $w = 10110$, ποια κωδικολέξη είναι πιο πιθανή να έχει αποσταλεί;

1.6.8 Επαναλάβετε την άσκηση 1.6.3, αφού αντικαταστήσετε τον C με $\{010101, 110110, 101101, 100110, 011001\}$ και τη w με 101010.

1.6.9 Ποιες κωδικολέξεις από τις $110110, 110101, 000111, 100111, 101000$ είναι πιο πιθανές να έχουν αποσταλεί, αν λάβαμε την $w = 011001$;

1.6.10 Στο θεώρημα 1.6.3 θεωρούμε ότι $1/2 < p < 1$. Τι θα μπορούσαμε ν' αλλάξουμε στην πρόταση του θεωρήματος 1.6.3, εάν αλλάξουμε τις υποθέσεις με:

(α'). $0 < p < 1/2$,

(β'). $p = 1/2$;

1.7 Ολίγη Βασική Άλγεβρα

Ένα πρόβλημα που θα στρέψουμε την προσοχή μας, είναι να βρούμε έναν αποτελεσματικό τρόπο να βρίσκουμε την κοντινότερη κωδικολέξη σε μια παραληφθείσα λέξη. Εάν ο κώδικας μας έχει πάρα πολλές κωδικολέξεις, τότε είναι πρακτικά άσκοπο να συγκρίνουμε κάθε φορά μια παραληφθείσα λέξη w με κάθε κωδικολέξη, για να βρούμε ποια κωδικολέξη συμφωνεί περισσότερο με τη w.

Για παράδειγμα, εάν ο κώδικας περιέχει 2^{12} κωδικολέξεις (όπως χρησιμοποιήθηκε στην αποστολή του Voyager) τότε σε μια τέτοια διαδικασία αποκωδικοποίησης, δε θα ελπίζαμε ποτέ να συγχρονιστούμε με το εισερχόμενο μήνυμα. Για να ξεπεράσουμε αυτό το πρόβλημα, εισάγουμε κάποια δομή στους κώδικες.

Έστω $K = \{0, 1\}$ και έστω K^n να είναι το σύνολο όλων των δυαδικών λέξεων με μήκος n. Ορίζουμε τη δυαδική πρόσθεση και τον δυαδικό πολλαπλασιασμό κατά τα γνωστά:

$$0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 0$$

$$0 \cdot 0 = 0, 1 \cdot 0 = 0, 0 \cdot 1 = 0, 1 \cdot 1 = 1$$

Ορίζουμε την πρόσθεση για τα στοιχεία του K^n κατά συντεταγμένες, χρησιμοποιώντας τη δυαδική πρόσθεση του K για κάθε θέση. Για παράδειγμα, έστω

$$v = 01101 \text{ και } w = 11001 \text{ τότε } v + w = 10100.$$

Προφανώς η πρόσθεση δύο δυαδικών λέξεων μήκους n έχει ως αποτέλεσμα μια δυαδική λέξη μήκους n, οπότε το K^n είναι κλειστό ως προς την πρόσθεση.

Χρησιμοποιώντας την ορολογία της Γραμμικής Άλγεβρας, τα στοιχεία 0 και 1 του K λέγονται βαθμωτά. Τότε ο βαθμωτός πολλαπλασιασμός στο K^n ορίζεται κατά συντεταγμένες. Επειδή τα μοναδικά βαθμωτά μεγέθη είναι το 0 και το 1, τα μοναδικά βαθμωτά πολλαπλάσια μιας λέξης w είναι το $0 \cdot w$, που είναι το στοιχείο του K^n με 0 σε κάθε συντεταγμένη και το $1 \cdot w$, που είναι η w. Ορίζουμε το στοιχείο του K^n με 0 σε όλες τις συντεταγμένες ως την *μηδενική λέξη*. Προφανώς το K^n είναι κλειστό ως προς βαθμωτό πολλαπλασιασμό.

Με αυτούς τους ορισμούς της πρόσθεσης και του βαθμωτού πολλαπλασιασμού, μπορεί εύκολα να δειχθεί ότι το K^n είναι διανυσματικός χώρος ως προς K. Δηλαδή, για οποιαδήποτε λέξεις n, u, v και w και για κάθε βαθμωτά a και b:

1. $v + w \in K^n$
2. $(u + v) + w = u + (v + w)$
3. $v + 0 = 0 + v = v$, όπου με 0 συμβολίζουμε τη μηδενική λέξη
4. για κάποια $v' \in K^n, v + v' = v' + v = 0$
5. $v + w = w + v$
6. $av \in K^n$
7. $a(v + w) = av + aw$
8. $(a + b)v = av + bv$
9. $(ab)v = a(bv)$
10. $1v = v$.

Ασκήσεις

1.7.1 Δείξτε ότι αν v είναι μια λέξη του K^n, τότε $v + v = 0$.

1.7.2 Δείξτε ότι αν v και w είναι λέξεις του K^n και $v + w = 0$, τότε $v = w$.

1.7.3 Δείξτε ότι αν v, u και w είναι λέξεις του K^n και $u + v = w$, τότε και $u + w = v$.

Ας παρατηρήσουμε ότι αν v είναι η κωδικολέξη που αποστέλνεται μέσω ενός ΔΣΚ και w η λέξη που λαμβάνεται, τότε εάν κάποια συντεταγμένη (ψηφίο) του w μεταδόθηκε εσφαλμένα, τότε η $v + w$ περιέχει 1 σε αυτή τη θέση, ενώ εάν μεταδόθηκε σωστά, τότε περιέχει 0. Το $v+w$ ονομάζεται *υπόδειγμα λάθους (pattern error),* ή σκέτο *λάθος*. Για παράδειγμα, εάν αποστέλλεται η $v = 10101$ και λαμβάνεται η $w = 01101$, τότε εμφανίζονται λάθη στην 1η, 2η και 5η θέση. Το υπόδειγμα λάθους είναι $v + w = 11001$.

1.8 Βάρος και Απόσταση

Θα εισάγουμε δύο σημαντικούς ορισμούς. Έστω v μια λέξη μήκους n. Το *βάρος Hamming,* ή απλά το *βάρος* της v, είναι το πλήθος των εμφανίσεων του ψηφίου 1 στη v. Συμβολίζουμε το βάρος της v με $wt(v)$. Για παράδειγμα, $wt(110101) = 4$ και $wt(00000) = 0$.

Έστω v και w δύο λέξεις μήκους n. Η *απόσταση Hamming,* ή απλά *απόσταση* μεταξύ των v και w, είναι το πλήθος των θέσεων στις οποίες οι v και w διαφωνούν. Συμβολίζουμε την απόσταση μεταξύ των v και w με $d(v, w)$. Για παράδειγμα, $d(01011, 00111) = 2$ και $d(10110, 10110) = 0$.

Παρατηρήστε ότι η απόσταση μεταξύ των v και w, είναι η ίδια με το βάρος του υποδείγματος λάθους $u = v + w$:

$$d(v, w) = wt(v + w).$$

Για παράδειγμα, εάν $v = 11010$ και $w = 01101$, τότε έχουμε

$d(v, w) = d(11010, 01101) = 4$ και $wt(u+w) = wt(11010+01101) = wt(10111) = 4$.

Έτσι ο τύπος της πιθανότητας στην ενότητα 1.6, μπορεί να ξαναεκφραστεί ως:

$$\phi_p(v, w) = p^{n-wt(u)}(1 - p)^{wt(u)},$$

όπου u είναι το υπόδειγμα λάθους $u = v + w$. Θα αναφερόμαστε στο $\phi_p(v, w)$ ως *την πιθανότητα του υποδείγματος λάθους* $u = v + w$.

Ασκήσεις

1.8.1 Υπολογίστε το βάρος της καθεμιάς από τις παρακάτω λέξεις και την απόσταση μεταξύ των ζευγαριών: $u_1 = 1001010$, $u_2 = 0110101$, $u_3 = 0011110$, και $u_4 = u_2 + u_3$.

1.8.2 Έστω $u = 01011$, $v = 11010$, $w = 01100$. Συγκρίνετε καθένα από τα παρακάτω ζευγάρια ποσοτήτων:

(α'). $wt(v + w)$ και $wt(v) + wt(w)$,

(β'). $d(v, w)$ και $d(v, u) + d(u, w)$.

Παρακάτω απαριθμούμε έναν αριθμό από σχέσεις που αφορούν το βάρος και την απόσταση. Εδώ u, v και w είναι λέξεις μήκους n και a είναι ένα ψηφίο.

1. $0 \leq wt(v) \leq n$
2. $wt(0) = 0$
3. Εάν $wt(v) = 0$, τότε $v = 0$.
4. $0 \leq d(v, w) \leq n$
5. $d(v, v) = 0$
6. Εάν $d(v, w) = 0$, τότε $v = w$.
7. $d(v, w) = d(w, v)$
8. $wt(v + w) \leq wt(v) + wt(w)$
9. $d(v, w) \leq d(v, u) + d(u, w)$
10. $wt(av) = a \cdot wt(v)$
11. $d(av, aw) = a \cdot d(v, w)$.

Οι περισσότερες από αυτές τις σχέσεις είναι άμεσα προφανείς από τους ορισμούς του βάρους και της απόστασης. Στην άσκηση 1.8.2, ο αναγνώστης κατασκεύασε παραδείγματα των σχέσεων 8 και 9. Για να κατασκευάσετε αποδείξεις, προσπαθείστε να χρησιμοποιήσετε τη βασική σχέση $d(v, w) = wt(v + w)$ και τις Ασκήσεις 1.7.1, 1.7.2 και 1.7.3 κατάλληλα.

Ασκήσεις

1.8.3 Κατασκευάστε ένα παράδειγμα στον K^5 για καθένα από τους έντεκα παραπάνω κανόνες.

1.8.4 Αποδείξτε τους έντεκα παραπάνω κανόνες.

Αυτές οι σχέσεις θα χρησιμοποιούνται όταν χρειαστεί και χωρίς κάποιο σχόλιο στις επόμενες ενότητες.

1.9 Αποκωδικοποίηση Μέγιστης Πιθανότητας

Τώρα είμαστε έτοιμοι να δώσουμε πιο ακριβείς τυποποιήσεις δύο βασικών προβλημάτων της θεωρίας κωδίκων. Ας υποθέσουμε ότι είμαστε στο ένα άκρο ενός ΔΣΚ και θέλουμε να λάβουμε ένα μήνυμα από τον μεταδότη (πομπό), που βρίσκεται στην άλλη άκρη του καναλιού. Ο μεταδότης είναι φυσικά κάποιος που εμείς έχουμε σχεδιάσει προηγουμένως. Πράγματι, ο σχεδιασμός ενός καλού μεταδότη είναι ένα από τα βασικά προβλήματα.

Υπάρχουν δύο ποσότητες που δεν μπορούμε να ελέγξουμε. Η μία είναι η πιθανότητα p, που το ΔΣΚ θα μεταδώσει κάποιο ψηφίο σωστά. Η δεύτερη είναι το πλήθος των δυνατών μηνυμάτων που μπορούν να μεταδοθούν. Τα πραγματικά μηνύματα που αποστέλλονται δεν είναι τόσο σημαντικά, όσο σημαντικό είναι το πλήθος των δυνατών μηνυμάτων. Για παράδειγμα, μόνο δύο μηνύματα ήταν απαραίτητα, έως ότου ο Paul Revere ξεκινήσει τον περίφημο μεταμεσονύκτιο περίπατο.

Ας υποθέσουμε ότι για κάθε σύνολο S με $|S|$ συμβολίζουμε το *πλήθος των στοιχείων* του S. Έτσι $|K^n| = 2^n$ από την άσκηση 1.2.2.

Τα δύο βασικά προβλήματα στην κωδικοποίηση είναι:

1.9.1 Κωδικοποίηση Πρέπει να ορίσουμε τον κώδικα που θα χρησιμοποιήσουμε για να αποστείλουμε τα μηνύματα. Πρέπει να κάνουμε κάποιες επιλογές. Κατ'αρχάς, θα επιλέξουμε ένα θετικό ακέραιο k, το μήκος κάθε δυαδικής λέξης που αντιστοιχεί σ' ένα μήνυμα. Επειδή κάθε μήνυμα πρέπει να αντιστοιχεί σε μια διαφορετική δυαδική λέξη μήκους k, το k πρέπει να επιλεχθεί έτσι ώστε $|M| \leq |K^k| = 2^k$. Στη συνέχεια πρέπει να αποφασίσουμε πόσα επιπλέον ψηφία είναι ανάγκη να προσθέσουμε σε κάθε λέξη μήκους k, ώστε να εξασφαλίσουμε την ανίχνευση ή τη διόρθωση, όσο το δυνατόν περισσότερων λαθών, δηλαδή ποιες θα είναι οι κωδικολέξεις και ποιο το μήκος n του κώδικα. Για να μεταδοθεί ένα συγκεκριμένο μήνυμα, ο μεταδότης βρίσκει κατ'αρχάς τη λέξη μήκους k που αντιστοιχεί στο μήνυμα και στη συνέχεια μεταδίδει την κωδικολέξη που αντιστοιχεί σ' αυτήν τη λέξη.

1.9.2 Αποκωδικοποίηση Έστω ότι μια λέξη w του K^n λαμβάνεται. Θα περιγράψουμε τώρα

έναν αλγόριθμο που λέγεται *Αποκωδικοποίηση Μέγιστης Πιθανότητας (Maximum Likelihood Decoding),* ή *ΑΜΠ* για να αποφασίσουμε ποια λέξη v του κώδικα C αποστάλθηκε. Υπάρχουν ουσιαστικά δύο είδη της ΑΜΠ:

1) *Πλήρης Αποκωδικοποίηση Μέγιστης Πιθανότητας (Complete Maximum Likelihood Decoding),* ή *ΠΑΜΠ.* Εάν υπάρχει μια μοναδική λέξη v του C πιο κοντά στη w από κάθε άλλη λέξη στο C, τότε αποκωδικοποιούμε την w ως v. Δηλαδή, αν $d(v, w) < d(v_1, w)$ για κάθε v_1 στο C, με $v_1 \neq v$, τότε αποκωδικοποιούμε w ως v. Εάν υπάρχουν πολλές λέξεις του C πιο κοντά στη w, δηλαδή στην ίδια απόσταση από το w, τότε επιλέγουμε αυθαίρετα κάποια απ' αυτές και συμπεραίνουμε ότι αυτή είναι η κωδικολέξη που στάλθηκε.

2) *Ημιτελής Αποκωδικοποίηση Μέγιστης Πιθανότητας (Incomplete Maximum Likelihood Decoding),* ή *ΗΑΜΠ.* Ξανά, εάν υπάρχει μοναδική λέξη v στο C

πιο κοντά στο w, τότε αποκωδικοποιούμε το w ως v. Εάν όμως υπάρχουν πολλές λέξεις στο C στην ίδια απόσταση από την w, τότε ζητάμε επανάληψη του μηνύματος, όπως επίσης και σε μερικές περιπτώσεις που η ληφθείσα λέξη w είναι πολύ μακριά από κάθε λέξη του κώδικά μας.

Θα χρησιμοποιήσουμε την ΗΑΜΠ για τα παραδείγματα και τις ασκήσεις της ενότητας, αλλά και στο μεγαλύτερο μέρος του υπόλοιπου κειμένου. Τονίζουμε ότι η ΑΜΠ δε δουλεύει πάντα. Συγκεκριμένα, αν έχουν γίνει πάρα πολλά λάθη κατά τη μετάδοση του μηνύματος μέσα από το ΔΣΚ η ΑΜΠ αποτυγχάνει.

Η λέξη v του C που είναι πιο κοντά στη ληφθείσα λέξη w, είναι εκείνη για την οποία η απόσταση $d(v,w)$ είναι ελάχιστη και συνεπώς από το θεώρημα 1.6.3, έχει τη μέγιστη πιθανότητα $\phi_p(v,w)$ για να είναι η λέξη που πραγματικά στάλθηκε. Το παράδειγμα 1.6.4 αποδεικνύει το γεγονός αυτό. Επειδή $d(v,w) = wt(v+w)$, το βάρος του υποδείγματος λάθους $u = v + w$, το θεώρημα 1.6.3 μπορεί να περιγραφεί ως εξής:

$$\phi_p(v_1,w) \leq \phi_p(v_2,w) \Leftrightarrow wt(v_1+w) \geq wt(v_2+w)\cdot$$

δηλαδή, *η πιο πιθανή αποσταλείσα λέξη είναι εκείνη με το μικρότερο βάρος υποδείγματος λάθους*.

Έτσι η στρατηγική στην ΗΑΜΠ είναι να εξετάσουμε τα υποδείγματα λάθους $v + w$ για όλες τις κωδικολέξεις v και να επιλέξουμε τη v η οποία παράγει το μικρότερο βάρος στο υπόδειγμα λάθους.

Παράδειγμα 1.9.3 Υποθέστε ότι $|M| = 2$ και επιλέγουμε $h = 3$ και $C = \{000, 111\}$. Εάν $n = 000$ αποστέλλεται, τότε πότε η ΗΑΜΠ θα αποφασίσει σωστά ότι στάλθηκε η v; Επίσης, πότε η ΗΑΜΠ θα αποφασίσει λάθος ότι στάλθηκε η 111; Κατασκευάζουμε τον Πίνακα 1.1 όπως ακολουθεί.

Ληφθείσα	Υπόδειγμα Λάθους		Αποκωδικοποίηση
w	$000+w$	$111+w$	v
000	000∗	111	000
100	100∗	011	000
010	010∗	101	000
001	001∗	110	000
110	110	001∗	111
101	101	010∗	111
011	011	100∗	111
111	111	000∗	111

Πίνακας 1.1: ΗΑΜΠ πίνακας για το παράδειγμα 1.9.3

Η πρώτη στήλη περιέχει όλες τις δυνατές λέξεις που μπορεί να παραληφθούν, δηλαδή όλο το K^3. Η δεύτερη και η τρίτη περιέχουν τα υποδείγματα λάθους $v+w$ για κάθε λέξη v του κώδικα C. Επειδή η ΗΑΜΠ θα επιλέγει τα υποδείγματα

λάθους με το μικρότερο βάρος, θέσαμε έναν αστερίσκο δίπλα από κάθε λέξη ελάχιστου βάρους. Στην τελευταία στήλη γράψαμε τη λέξη v του κώδικα C, που αντιστοιχεί στη στήλη όπου ο αστερίσκος έχει τοποθετηθεί. Αυτή είναι η λέξη v που η ΗΑΜΠ θα υποθέσει ότι στάλθηκε, για κάθε λέξη που λαμβάνεται. Έτσι η ΗΑΜΠ θα συμπεράνει σωστά ότι στάλθηκε 000 αν παραλήφθηκε κάποια από τις 000, 100, 010 ή 001 (τέσσερις πρώτες γραμμές του πίνακα 1.1). Ενώ η ΗΑΜΠ θα συμπεράνει εσφαλμένα ότι στάλθηκε 111, εάν παραλήφθηκε κάποια από τις 110, 101, 011 ή 111.

Παράδειγμα 1.9.4 Υποθέστε ότι $|M| = 3$ και επιλέγουμε $C = \{0000, 1010, 0111\}$ με $h = 4$. Θα φτιάξουμε τον ΗΑΜΠ Πίνακα 1.2, όπως κάναμε και στο προηγούμενο παράδειγμα, εκτός εάν δύο ή περισσότερες θέσεις στις στήλες των υποδειγμάτων λαθών έχουν το ίδιο ελάχιστο βάρος, οπότε δε θέτουμε αστερίσκο σ' αυτήν τη γραμμή και δε γράφουμε τίποτα (δηλώνεται με -) στη στήλη του v με την αποκωδικοποιημένη λέξη. Αυτό σημαίνει για την ΗΑΜΠ, ότι απαιτούμε αναμετάδοση κάθε φορά που σε δύο ή περισσότερες λέξεις το υπόδειγμα λάθους έχει το ίδιο ελάχιστο βάρος.

Ληφθείσα	Υπόδειγμα Λάθους			Αποκωδικοποίηση
w	$0000 + w$	$1010 + w$	$0111 + w$	v
0000	0000∗	1010	0111	0000
1000	1000	0010	1111	–
0100	0100∗	1110	0011	0000
0010	0010	1000	0101	–
0001	0001∗	1011	0110	0000
1100	1100	0110	1011	–
1010	1010	0000∗	1101	1010
1001	1001	0011	1110	–
0110	0110	1100	0001∗	0111
0101	0101	1111	0010∗	0111
0011	0011	1001	0100∗	0111
1110	1110	0100∗	1001	1010
1101	1101	0111	1010∗	0111
1011	1011	0001∗	1100	1010
0111	0111	1101	0000∗	0111
1111	1111	0101	1000∗	0111

Πίνακας 1.2: ΗΑΜΠ πίνακας για το παράδειγμα 1.9.4

Ασκήσεις

1.9.5 Έστω $|M| = 2$, $n = 3$ και $C = \{001, 101\}$. Εάν σταλεί $v = 001$, πότε η ΗΑΜΠ θα αποφασίσει ότι στάλθηκε αυτό σωστά και πότε η ΗΑΜΠ θα αποφασίσει ότι στάλθηκε το 101 λανθασμένα;

1.9.6 Έστω $|M| = 3$ και $n = 3$. Για κάθε λέξη w στο K^3 που λαμβάνουμε, βρείτε τη λέξη v στον κώδικα $C = \{000, 001, 110\}$ την οποία η ΗΑΜΠ θα αποφασίσει ότι έχει σταλεί.

1.9.7 Κατασκευάστε έναν ΗΑΜΠ πίνακα για καθέναν από τους παρακάτω κώδικες:

(α'). $C = \{101, 111, 011\}$

(β'). $C = \{000, 001, 010, 011\}$

(γ'). $C = \{0000, 0001, 1110\}$

(δ'). $C = \{0000, 1001, 0110, 1111\}$

(ε'). $C = \{00000, 11111\}$

(ϛ'). $C = \{00000, 11100, 00111, 11011\}$

(ζ'). $C = \{00000, 11110, 01111, 10001\}$

(η'). $C = \{000000, 101010, 010101, 111111\}$

Υπενθυμίζουμε ότι πρέπει να επιλέγουμε n και C (1.9.1). Μερικές επιλογές είναι καλύτερες από άλλες. Παραθέτουμε τρία σημαντικά κριτήρια για σύγκριση καλών επιλογών:

1) Μεγαλύτερες λέξεις απαιτούν περισσότερο χρόνο μετάδοσης και αποκωδικοποίησης, ώστε το n δε θα πρέπει να είναι πολύ μεγάλο, δηλαδή ο βαθμός πληροφορίας πρέπει να είναι όσο το δυνατόν πιο κοντά στη 1.

2) Εάν το $|C|$ είναι μεγάλο - ας πούμε κάμποσες χιλιάδες ή και μεγαλύτερο - και στέλνονται πολλά μηνύματα ανά δευτερόλεπτο, τότε ο αλγόριθμος για την ΗΑΜΠ που περιγράφηκε σ' αυτήν την ενότητα, θα ξοδέψει πολύ χρόνο για να εφαρμοστεί. Ευτυχώς, κάποιες έξυπνες επιλογές του C επιτρέπουν πιο επιτυχημένες και γρήγορες μεθόδους για την ΗΑΜΠ.

3) Εάν έχουν συμβεί πολλά λάθη κατά τη μετάδοση, τότε η ΑΜΠ δε θα δουλέψει. Δηλαδή, η λέξη εκείνη που η ΑΜΠ θα διαλέξει ότι δήθεν αποστάλθηκε, δε θα είναι η ίδια με την πραγματική. Οπότε, ο C θα πρέπει να επιλεχτεί, ώστε η πιθανότητα η ΑΜΠ να δουλέψει, να είναι πολύ μεγάλη. (Θεωρούμε αυτήν την πιθανότητα στην επόμενη ενότητα).

Έτσι, ισχυριζόμαστε ότι *ο βασικός σκοπός της θεωρίας κωδίκων είναι να βρει σύνολα C λέξεων τα οποία, σύμφωνα με τα παραπάνω κριτήρια, να είναι ικανοποιητικά.* Οι περισσότερες από τις προσπάθειές μας θα έχουν αυτό το σκοπό.

1.10 Αξιοπιστία της ΑΜΠ

Υποθέτουμε ότι το n και το C έχουν επιλεγεί. Τώρα θα δώσουμε μια διαδικασία (έναν αλγόριθμο) για να προσδιορίσουμε την πιθανότητα $\theta_p(C, v)$, για κάθε αποσταλλείσα λέξη v μέσω ενός ΔΣΚ με πιθανότητα p, που η ΗΑΜΠ σωστά θα αποφασίσει ότι στάλθηκε η v.

Κατ'αρχάς, βρίσκουμε το σύνολο $L(v)$ των λέξεων του K^n που είναι πιο κοντά στη v, από κάθε άλλη λέξη στο C. Τότε ορίζουμε το $\theta_p(C, v)$ να είναι το άθροισμα όλων των πιθανοτήτων $\phi_p(v, w)$, καθώς το w διατρέχει το $L(v)$. Δηλαδή:

$$\theta_p(C, v) = \sum_{w \in L(v)} \phi_p(v, w).$$

Σημειώστε ότι το $L(v)$ είναι το σύνολο των λέξεων του K^n, οι οποίες, εάν παραλειφθούν, η ΗΑΜΠ θα συμπεράνει ορθά ότι η v έχει αποσταλεί. Μπορούμε να βρούμε το $L(v)$ από τον ΗΑΜΠ πίνακα που κατασκευάστηκε στην τελευταία ενότητα. Σε κάθε γραμμή του πίνακα, η λέξη w στην πρώτη στήλη ανήκει στο $L(v)$, αν στην τελευταία στήλη και στην ίδια γραμμή υπάρχει το v και αυτές είναι και όλες οι λέξεις του $L(v)$.

Παρατηρήστε επίσης ότι το $\theta_p(C, v)$ είναι το άθροισμα πάνω στις λέξεις w του $L(v)$ των πιθανοτήτων των υποδειγμάτων λάθους $v + w$ που συμβαίνουν κατά τη μετάδοση.

Το θ_p μπορεί να χρησιμοποιηθεί για να συγκριθούν δύο κώδικες, κρίνοντας τους με το τρίτο κριτήριο της προηγούμενης παραγράφου. Όμως, θα πρέπει να παρατηρήσουμε ότι το $\theta_p(C, v)$, έτσι όπως ορίστηκε, δεν λαμβάνει υπόψη την περίπτωση της αναμετάδοσης, όταν η παραληφθείσα λέξη ισαπέχει από δύο κωδικολέξεις. Αυτό οδηγεί σε κάποιες ανωμαλίες (όπως $\theta_p(K^n, v) > \theta_p(C, v)$ για κάθε v στο K^n και u στο C, όπου C είναι ο parity check κώδικας που δημιουργείται από το K^n), αλλά είναι μια αποδεκτή προσέγγιση του μέτρου της αξιοπιστίας. Βεβαίως, το $\theta_p(C, v)$ είναι το κάτω φράγμα της πιθανότητας που η v αποκωδικοποιείται ορθά.

Παράδειγμα 1.10.1 Υποθέτουμε ότι $p = 0.90, |M| = 2, u = 3$ και $C = \{000, 111\}$, όπως στο παράδειγμα 1.9.3. Εάν η λέξη $v = 000$ έχει σταλεί, θα υπολογίσουμε την πιθανότητα που η ΗΑΜΠ θα συμπεράνει σωστά αυτήν, μετά την αποστολή της. Από τον πίνακα 1.1, η $v = 000$ αποκωδικοποιείται στις πρώτες τέσσερις γραμμές, ώστε το σύνολο $L(000)$ (των λέξεων του K^3 πιο κοντά στην $v = 000$ παρά στην 111) είναι:

$$L(000) = \{000, 100, 010, 001\}.$$

Έτσι,

$$\begin{aligned}
\theta_p(C, 000) &= \theta_p(000, 000) + \theta_p(000, 100) + \theta_p(000, 010) + \theta_p(000, 001) \\
&= p^3 + p^2(1-p) + p^2(1-p) + p^2(1-p) \\
&= p^3 + 3p^2(1-p) \\
&= 0.972 \text{ (υποθέτοντας ότι } p = 0.9).
\end{aligned}$$

Εάν η $v = 111$ μεταδόθηκε, υπολογίζουμε την πιθανότητα που η ΗΑΜΠ θα συμπεράνει σωστά αυτή, μετά από μια αποστολή της. Κατ'αρχάς,

$$L(111) = \{110, 101, 011, 111\},$$

οπότε

$$\begin{aligned}
\theta_p(C, 111) &= \theta_p(111, 110) + \theta_p(111, 101) + \theta_p(111, 011) + \theta_p(111, 111) \\
&= p^2(1-p) + p^2(1-p) + p^2(1-p) + p^3 \\
&= 3p^2(1-p) + p^3 \\
&= 0,972 \text{ (υποθέτοντας ότι } p = 0.9).
\end{aligned}$$

Ασκήσεις

1.10.2 Έστω ότι $p = 0.90$, $|M| = 2$, $n = 3$ και $C = \{001, 101\}$, όπως στην άσκηση 1.9.5.

(α'). Εάν η λέξη $v = 001$ έχει σταλεί, βρείτε την πιθανότητα που η ΗΑΜΠ θα συμπεράνει σωστά αυτή μετά από μια αποστολή της.

(β'). Επαναλάβετε το (α') ερώτημα για $v = 101$.

Οι απαντήσεις και στα δύο ερωτήματα στην παραπάνω άσκηση είναι $\theta_p(C, v) = 0.900$. Συγκρίνοντάς αυτές με τα αποτελέσματα του παραδείγματος 1.10.1, συμπεραίνουμε ότι αφού $0.900 < 0.972$, ο κώδικας $C = \{000, 111\}$ είναι καλύτερος από τον $C = \{001, 101\}$, αν τουλάχιστον αποφασίσουμε σύμφωνα με το τρίτο κριτήριο της τελευταίας παραγράφου. Η μέθοδος μας δίνει έναν αλγόριθμο (αν και αποτελεσματικός για μεγάλο n) για να αποφασίσουμε αν η πιθανότητα να δουλέψει η ΗΑΜΠ είναι υψηλή. Ευτυχώς, οι περισσότεροι από τους κώδικες που αργότερα θα σχεδιάσουμε, έχουν τέτοια δομή, ώστε ο υπολογισμός της πιθανότητας, είναι πολύ πιο εύκολος.

Παράδειγμα 1.10.3 Υποθέστε ότι $p = 0.90$, $|M| = 3$, $n = 4$ και $C = \{0000, 1010, 0111\}$, όπως στο παράδειγμα 1.9.4. Για κάθε v στο C, υπολογίζουμε τη $\theta_p(C, v)$.

(α').

$$\begin{aligned}
v &= 0000 \\
L(0000) &= \{0000, 0100, 0001\} \quad \text{(από τον πίνακα 1.2)} \\
\theta_p(C, v) &= \theta_p(0000, 0000) + \theta_p(0000, 0100) + \theta_p(0000, 0001) \\
&= p^4 + p^3(1-p) + p^3(1-p) \\
&= p^4 + 2p^3(1-p) = 0,8019
\end{aligned}$$

(β').

$$\begin{aligned} v &= 1010 \\ L(1010) &= \{1010, 1110, 1011\} \\ \theta_p(C, v) &= \theta_p(1010, 1010) + \theta_p(1010, 1110) + \theta_p(1010, 1011) \\ &= p^4 + p^3(1-p) + p^3(1-p) \\ &= p^4 + 2p^3(1-p) = 0,8019 \end{aligned}$$

(γ').

$$\begin{aligned} v &= 0111 \\ L(0111) &= \{0110, 0101, 0011, 1101, 0111, 1111\} \\ \theta_p(C, v) &= \theta_p(0111, 0110) + \theta_p(0111, 0101) + \theta_p(0111, 0011) \\ &\quad +\theta_p(0111, 1101) + \theta_p(0111, 0111) + \theta_p(0111, 1111) \\ &= p^3(1-p) + p^3(1-p) + p^3(1-p) + p^2(1-p)^2 + p^4 + p^3(1-p) \\ &= p^4 + 4p^3(1-p) + +p^2(1-p)^2 = 0.9588 \end{aligned}$$

Εξετάζοντας τις τρεις πιθανότητες που υπολογίσαμε, βλέπουμε ότι η πιθανότητα, ώστε η ΗΑΜΠ να συμπεράνει σωστά ότι έχει αποσταλεί η 0111, δεν είναι τόσο κακή. Ωστόσο, η πιθανότητα να συμπεράνει σωστά η ΗΑΜΠ ότι έχει αποσταλεί η 0000 ή η 1010 είναι απαράδεκτη. Έτσι, τουλάχιστον ως προς το τρίτο κριτήριο της τελευταίας παραγράφου, ο $C = \{0000, 1010, 0111\}$ δεν είναι ιδιαίτερα καλή επιλογή για έναν κώδικα.

Ασκήσεις

1.10.4 Υποθέτουμε ότι $p = 0.90$ και $C = \{000, 001, 110\}$ όπως στην άσκηση 1.9.6. Εάν η λέξη $v = 110$ έχει σταλεί, να υπολογίσετε την πιθανότητα που η ΗΑΜΠ θα συμπεράνει σωστά αυτή και την πιθανότητα που η ΗΑΜΠ δε θα συμπεράνει σωστά ότι στάλθηκε η λέξη 000.

1.10.5 Για κάθε έναν από τους παρακάτω κώδικες C, υπολογίστε την $\theta_p(C, v)$, χρησιμοποιώντας $p = 0.90$. (Οι πίνακες της ΗΑΜΠ γι' αυτούς τους κώδικες, έχουν κατασκευαστεί στην άσκηση 1.9.7)

(α'). $C = \{101, 111, 011\}$

(β'). $C = \{000, 001, 010, 011\}$

(γ'). $C = \{0000, 0001, 1110\}$

(δ'). $C = \{0000, 1001, 0110, 1111\}$

(ε'). $C = \{00000, 11111\}$

(ϛ΄). $C = \{00000, 11100, 00111, 11011\}$

(ζ΄). $C = \{00000, 11110, 01111, 10001\}$

(η΄). $C = \{000000, 101010, 010101, 111111\}$

1.11 Κώδικες ανίχνευσης λαθών

Τώρα θα δώσουμε αυστηρά την έννοια πότε ένας κώδικας C θα ανιχνεύει λάθη. Υπενθυμίζουμε ότι εάν η v στο C έχει αποσταλεί και η w στο K^n έχει ληφθεί, τότε η $u = v+w$ είναι το υπόδειγμα λάθους. Κάθε λέξη u στο K^n μπορεί να εμφανιστεί ως ένα υπόδειγμα λάθους και θέλουμε να ξέρουμε ποια υποδείγματα λάθους το C θα ανιχνεύσει.

Λέμε ότι ο κώδικας C *ανιχνεύει* ένα υπόδειγμα λάθους u, αν και μόνο αν το $v + u$ δεν είναι κωδικολέξη, για κάθε v στο C. Με άλλα λόγια, η u ανιχνεύεται, αν για κάθε αποσταλείσα κωδικολέξη v, ο αποκωδικοποιητής, όταν θα λάβει την $v+u$, θα αναγνωρίσει ότι δεν είναι κωδικολέξη και συνεπώς κάποιο λάθος υπάρχει.

Παράδειγμα 1.11.1 Έστω $C = \{001, 101, 110\}$. Για το υπόδειγμα λάθους $u = 010$, υπολογίζουμε το $v + 010$, για κάθε v στο C:

$$001 + 010 = 011, 101 + 010 = 111, 110 + 010 = 100.$$

Καμιά από τις λέξεις 011, 111 και 100, ανήκει στο C, οπότε ο C ανιχνεύει το υπόδειγμα λάθους 010. Αντίθετα, για το υπόδειγμα λάθους $u = 100$, βρίσκουμε

$$001 + 100 = 101, 101 + 100 = 001, 110 + 100 = 010.$$

Επειδή τουλάχιστον ένα από τα παραπάνω αθροίσματα ανήκει στο C, ο C δεν ανιχνεύει το υπόδειγμα λάθους 100.

Ασκήσεις

1.11.2 Έστω κώδικας $C = \{001, 101, 110\}$. Αποφασίστε αν ο C θα ανιχνεύσει τα υποδείγματα λάθους (a) 011, (b) 001 και (g) 000.

1.11.3 Για καθέναν από τους παρακάτω κώδικες C, αποφασίστε αν ο C θα ανιχνεύσει ή όχι το u:

(α΄). $C = \{00000, 10101, 11100\}$

(i) $u = 10101$

(ii) $u = 01010$

(iii) $u = 11011$

(β΄). $C = \{1101, 0110, 1100\}$

(i) $u = 0010$

(ii) $u = 0011$

(iii) $u = 1010$

(γ'). $C = \{1000, 0100, 0010, 0001\}$

(i) $u = 1001$

(ii) $u = 1110$

(iii) $u = 0110$

1.11.4 Ποια υποδείγματα λάθους θα ανιχνεύσει ο κώδικας $C = K^n$;

1.11.5 (ι) Έστω C ο κώδικας που περιέχει τη μηδενική λέξη ως κωδικολέξη. Αποδείξτε ότι αν το υπόδειγμα λάθους u είναι μια κωδικολέξη, τότε ο C δε θα ανιχνεύσει το u.

(ιι) Αποδείξτε ότι κανένας κώδικας C δε θα ανιχνεύσει το μηδενικό υπόδειγμα λάθους $u = 0$.

Ο πίνακας που κατασκευάσαμε για την ΗΑΜΠ μπορεί να χρησιμοποιηθεί για να ορίσουμε ποιο υπόδειγμα λάθους ένας κώδικας C θα ανιχνεύσει. Η πρώτη λίστα περιέχει κάθε λέξη στο K^n. Έτσι, η πρώτη στήλη μπορεί να παρουσιαστεί ως η στήλη με όλα τα υποδείγματα λάθους w και συνεπώς οι στήλες με τα «υποδείγματα λάθους» στον ΗΑΜΠ πίνακα, περιέχουν τα αθροίσματα $v + w$, για όλα τα $v \in C$. Εάν σε κάποια συγκεκριμένη γραμμή, κανένα απ' αυτά τα αθροίσματα δεν είναι κωδικολέξη του C, τότε ο C ανιχνεύει το υπόδειγμα λάθους που βρίσκεται στην πρώτη στήλη αυτής της γραμμής.

Παράδειγμα 1.11.6 Θεωρούμε τον κώδικα $C = \{000, 111\}$ με ΗΑΜΠ πίνακα 1.1. Όλα τα πιθανά υποδείγματα λάθους βρίσκονται στην πρώτη στήλη. Για δοσμένη u, όλα τα αθροίσματα $v + u$, καθώς η v τρέχει στο C, εμφανίζονται στη δεύτερη και στην τρίτη στήλη της γραμμής που «ονοματίζει» η u. Εάν, καμιά από τις λέξεις δεν ανήκει στο C (δηλαδή καμιά δεν είναι η 000 ή η 111), τότε ο C ανιχνεύει τη u. Οπότε ο C ανιχνεύει τα υποδείγματα λάθους 100, 010, 001, 110, 101 και 011, όπως μπορούμε να δούμε θεωρώντας τις γραμμές 2 έως 7 του πίνακα αλλά δεν μπορεί να ανιχνεύσει τα υποδείγματα λάθους 000 και 111.

Ασκήσεις

1.11.7 Αποφασίστε αν τα υποδείγματα λάθους ανιχνεύονται από κάθε κώδικα της άσκησης 1.9.7, χρησιμοποιώντας τους ΗΑΜΠ πίνακες που έχουν κατασκευαστεί εκεί.

Μια διαφορετική και πιο γρήγορη μέθοδος για την εύρεση των υποδειγμάτων λάθους που ο κώδικας C μπορεί να ανιχνεύσει, είναι να βρούμε πρώτα όλα τα υποδείγματα λάθους που ο C δεν μπορεί να ανιχνεύσει. Τότε όλα τα υπόλοιπα υποδείγματα λάθους, μπορούν ν' ανιχνευτούν από τον C. Προφανώς, για κάθε ζευγάρι κωδικολέξεων v και w, εάν $e = v + w$, τότε το e δεν μπορεί να ανιχνευτεί

διότι $v+e=w$, που είναι μια κωδικολέξη. Άρα, το σύνολο όλων των υποδειγμάτων λάθους που δεν μπορούν να ανιχνευτούν από τον C, είναι το σύνολο όλων των λέξεων που μπορούν να γραφτούν ως το άθροισμα 2 κωδικολέξεων.

Παράδειγμα 1.11.8 Ας θεωρήσουμε τον κώδικα $\{000, 111\}$. Αφού

$$000 + 000 = 000, 000 + 111 = 111 \text{ και } 111 + 111 = 000,$$

το σύνολο των υποδειγμάτων λάθους που δεν μπορεί να ανιχνευτεί από τον C, είναι $\{000, 111\}$. Συνεπώς, όλα τα υποδείγματα λάθους στο $K^3\{000, 111\}$ μπορούν να ανιχνευθούν.

Παράδειγμα 1.11.9 Έστω $C = \{1000, 0100, 1111\}$. Αφού $1000 + 1000 = 0000$, $1000 + 0100 = 1100$, $1000 + 1111 = 0111$ και $0100 + 1111 = 1011$, το σύνολο των υποδειγμάτων λάθους που δεν μπορεί να ανιχνευτεί από τον C, είναι $\{0000, 1100, 0111, 1011\}$. Συνεπώς, όλα τα υποδείγματα λάθους στο $K^4\backslash\{0000, 1100, 0111, 1011\}$ μπορούν να ανιχνευθούν.

Ασκήσεις

1.11.10 Βρείτε τα υποδείγματα λάθους που ανιχνεύονται από κάθε κώδικα που ακολουθεί παρακάτω και συγκρίνετε τις απαντήσεις σας μ' αυτές της άσκησης 1.11.5.

(α΄). $C = \{101, 111, 011\}$

(β΄). $C = \{000, 001, 010, 011\}$

(γ΄). $C = \{0000, 0001, 1110\}$

(δ΄). $C = \{0000, 1001, 0110, 1111\}$

(ε΄). $C = \{00000, 11111\}$

(ϛ΄). $C = \{00000, 11100, 00111, 11011\}$

(ζ΄). $C = \{00000, 11110, 01111, 10001\}$

(η΄). $C = \{000000, 101010, 010101, 111111\}$

Υπάρχει επίσης ένας τρόπος να προσδιορίσουμε κάποια υποδείγματα λάθους που ο κώδικας C θα ανιχνεύσει, χωρίς κάποιο δικό μας έλεγχο. Κατ'αρχάς, πρέπει να εισάγουμε άλλο ένα νούμερο που συσχετίζεται με τον C.

Για έναν κώδικα C που περιέχει τουλάχιστον δύο λέξεις, η *απόσταση* του κώδικα C είναι ο μικρότερος από τους αριθμούς $d(v, w)$, καθώς οι v και w καθορίζουν όλες τις διαφορετικές τιμές του C. Σημειώστε ότι επειδή $d(v, w) = wt(v + w)$, η απόσταση του κώδικα είναι η μικρότερη τιμή του $wt(v+w)$, καθώς οι v και w, με $v \neq w$, διατρέχουν όλες τις πιθανές κωδικολέξεις.

Η απόσταση ενός κώδικα έχει πολλές από τις ιδιότητες της Ευκλείδειας απόστασης· αυτή η αντιστοιχία μπορεί να είναι χρήσιμη στην κατασκευή της έννοιας της απόστασης ενός κώδικα.

Παράδειγμα 1.11.11 Έστω $C = \{0000, 1010, 0111\}$. Τότε $d(0000, 1010) = 2$, $d(0000, 0111) = 3$ και $d(1010, 0111) = 3$. Έτσι, η απόσταση του C είναι 2.

Ασκήσεις

1.11.12 Βρείτε την απόσταση για καθέναν από τους παρακάτω κώδικες:

(α'). $C = \{101, 111, 011\}$

(β'). $C = \{000, 001, 010, 011\}$

(γ'). $C = \{0000, 0001, 1110\}$

(δ'). $C = \{0000, 1001, 0110, 1111\}$

(ε'). $C = \{00000, 11111\}$

(ϛ'). $C = \{00000, 11100, 00111, 11011\}$

(ζ'). $C = \{00000, 11110, 01111, 10001\}$

(η'). $C = \{000000, 101010, 010101, 111111\}$

1.11.13 Βρείτε την απόσταση του κώδικα που σχηματίζεται προσθέτοντας ένα parity check ψηφίο στον K^n.

Τώρα μπορούμε να δώσουμε ένα θεώρημα που βοηθάει στην ανίχνευση πολλών από τα υποδείγματα λάθους που ένας κώδικας θα ανιχνεύσει.

Θεώρημα 1.11.14 *Ένας κώδικας C με απόσταση d, θα ανιχνεύσει τουλάχιστον όλα τα μη μηδενικά υποδείγματα λάθους, βάρους μικρότερου ή ίσου με $d-1$. Επίσης, υπάρχει τουλάχιστον ένα υπόδειγμα λάθους, βάρους d, που ο κώδικας δε θα ανιχνεύσει.*

Σημείωση Παρατηρείστε ότι ο C μπορεί να ανιχνεύσει κάποια υποδείγματα λάθους βάρους d ή μεγαλύτερου, αλλά δεν ανιχνεύει όλα τα υποδείγματα λάθους βάρους d.

Απόδειξη: Έστω u ένα μη μηδενικό υπόδειγμα λάθους, με $wt(u) \leq d-1$ και έστω v στο C. Τότε

$$d(v, v+u) = wt(v+v+u) = wt(u) < d.$$

Αφού ο C έχει απόσταση d, $v+u$ δεν ανήκει στο C. Άρα ο C ανιχνεύει το u. Από τον ορισμό του d, υπάρχουν κωδικολέξεις v και u του C με $d(v,u) = d$. Ας θεωρήσουμε το υπόδειγμα λάθους $u = v + w$. Τώρα, το $w = v + u$ ανήκει στο C, οπότε ο C δεν μπορεί να ανιχνεύσει το υπόδειγμα λάθους u με βάρος d.

Ένας κώδικας είναι t κώδικας *ανίχνευσης λάθους*, εάν ανιχνεύει όλα τα υποδείγματα λάθους με βάρος το πολύ t και δεν ανιχνεύει τουλάχιστον ένα υπόδειγμα λάθους με βάρος $t+1$. Έτσι, από το θεώρημα 1.11.14, αν ένας κώδικας έχει απόσταση d, τότε είναι ένας $d-1$ κώδικας ανίχνευσης λάθους.

Παράδειγμα 1.11.15 Ο κώδικας $C = \{000, 111\}$ έχει απόσταση $d = 3$. Από το θεώρημα 1.11.14, ο C ανιχνεύει όλα τα υποδείγματα λάθους, βάρους 1 ή 2 και δεν ανιχνεύει το μοναδικό υπόδειγμα λάθους βάρους 3, δηλαδή το 111. Το μοναδικό υπόδειγμα λάθους που δεν περιλαμβάνει το θεώρημα 1.11.14 είναι το 000. Από την άσκηση 1.11.5 γνωρίζουμε ότι το 000 δεν μπορεί να ανιχνευθεί.

Το θεώρημα 1.11.14 δεν εμποδίζει έναν κώδικα C να ανιχνεύσει υπόδειγμα λάθους με βάρος d ή μεγαλύτερο. Πράγματι, ο C συνήθως ανιχνεύει και κάποια τέτοια υποδείγματα λάθους.

Παράδειγμα 1.11.16 Ο κώδικας $C = \{001, 101, 110\}$ έχει απόσταση $d = 1$. Όμως $d - 1 = 0$ και έτσι το θεώρημα 1.11.14 δε μας βοηθά να ορίσουμε ποια υποδείγματα λάθους θα ανιχνεύσει ο C. Όμως μας λέει ότι υπάρχει τουλάχιστον ένα υπόδειγμα λάθους, βάρους $d = 1$ που ο C δε θα ανιχνεύσει. Όπως είδαμε στο παράδειγμα 1.11.1, τέτοιο υπόδειγμα λάθους είναι το 100. Σημειώστε, όμως, ότι ο C θα ανιχνεύσει το υπόδειγμα λάθους 010 με $d = 1$.

Ασκήσεις

1.11.17 Ο κώδικας $C = \{0000, 1010, 0111\}$ έχει απόσταση $d = 2$. Χρησιμοποιώντας την άσκηση 1.11.5, δείξτε ότι το υπόδειγμα λάθους 1010 δεν ανιχνεύεται. Δείξτε ότι αυτό είναι το μοναδικό υπόδειγμα λάθους, βάρους 2, που ο C δεν ανιχνεύει. Βρείτε όλα τα υποδείγματα λάθους που ο C ανιχνεύει.

1.11.18 Βρείτε όλα τα υποδείγματα λάθους που ο κώδικας C_3 του παραδείγματος δεν ανιχνεύει. Σημειώστε ότι ο C_3 είναι ένας κώδικας ανίχνευσης ενός λάθους.

1.11.19 Για κάθε κώδικα C της άσκησης 1.11.12, βρείτε τα υποδείγματα λάθους τα οποία το θεώρημα 1.11.14 εγγυάται ότι ο C θα ανιχνεύσει.

1.11.20 Έστω ότι ο κώδικας C αποτελείται από όλες τις λέξεις μήκους 4 οι οποίες έχουν ίσο βάρος. Επίσης, βρείτε τα υποδείγματα λάθους που ο C ανιχνεύει.

1.12 Κώδικες διόρθωσης λαθών

Εάν μια λέξη v ενός κώδικα διαδίδεται μέσω ενός ΔΣΚ και εάν w λαμβάνεται, δημιουργώντας ένα υπόδειγμα λάθους $u = v + w$, τότε η ΗΑΜΠ θα συμπεράνει σωστά ότι η v έχει σταλεί, με την προϋπόθεση ότι η w είναι πιο κοντά στη v από κάθε άλλη κωδικολέξη. Εάν αυτό συμβαίνει κάθε φορά που εμφανίζεται ένα υπόδειγμα λάθους u, ανεξάρτητα από το ποια κωδικολέξη αποστέλλεται, τότε λέμε ότι ο C *διορθώνει* το υπόδειγμα λάθους u. Δηλαδή, λέμε ότι *ο κώδικας C διορθώνει το υπόδειγμα λάθους u, εάν για όλες τις v στο C, το $v + u$ είναι πιο κοντά στο v από κάθε άλλη λέξη στο C*. Επίσης, ένας κώδικας λέγεται t *κώδικας διόρθωσης*, αν διορθώνει όλα τα υποδείγματα λάθους, βάρους το πολύ t και δε διορθώνει τουλάχιστον ένα υπόδειγμα λάθους βάρους $t + 1$.

Παράδειγμα 1.12.1 Έστω $C = \{000, 111\}$.

(α΄). Πάρτε το υπόδειγμα λάθους $u = 010$. Για $v = 000$,

$$d(000, v + u) = d(000, 000) = 1 \text{ και}$$

$$d(111, v + u) = d(111, 010) = 2.$$

Και για $v = 111$,

$$d(111, v + u) = d(000, 101) = 2 \text{ και}$$

$$d(111, v + u) = d(111, 101) = 1.$$

Έτσι, ο C διορθώνει το υπόδειγμα λάθους $u = 010$.

(β΄). Ας πάρουμε τώρα το υπόδειγμα λάθους $u = 110$. Για $v = 000$,

$$d(000, v + u) = d(000, 110) = 2 \text{ και}$$

$$d(111, v + u) = d(111, 110) = 1.$$

Επειδή το $v + u$ δεν είναι πιο κοντά στο $v = 000$ παρά στο 111, ο C δε διορθώνει το υπόδειγμα λάθους 110.

Ο πίνακας της ΗΑΜΠ μπορεί να χρησιμοποιηθεί για να αποφασίσουμε ποιο υπόδειγμα λάθους ένας κώδικας C θα διορθώσει. Σε κάθε στήλη του πίνακα, όλα τα δυνατά υποδείγματα λάθους (που σημαίνει κάθε λέξη του K^n), εμφανίζονται μια μόνο φορά (διότι αλλιώς, αν το υπόδειγμα λάθους u εμφανίζεται σε κάποια στήλη δυο φορές για κάποια κωδικολέξη v, τότε το u εμφανίζεται σε διαφορετικές γραμμές που αντιστοιχούν σε διαφορετικές ληφθείσες λέξεις, ας πούμε w_1 και w_2· οπότε $u = v + w_1 = v + w_2$, το οποίο είναι άτοπο για $w_1 \neq w_2$). Επίσης, ένας αστερίσκος έχει τοποθετηθεί δίπλα στο υπόδειγμα λάθους u στη στήλη που αντιστοιχεί σε μια κωδικολέξη v στον ΗΑΜΠ πίνακα, ακριβώς όταν το $v + u$ είναι πιο κοντά στη v από κάθε άλλη λέξη. Άρα, ένα υπόδειγμα λάθους u διορθώνεται εάν ένας αστερίσκος τοποθετείται δίπλα στο u σε *κάθε* στήλη του ΗΑΜΠ πίνακα.

Παράδειγμα 1.12.2 Για τον κώδικα $C = \{000, 111\}$ ο ΗΑΜΠ πίνακας είναι ο 1.1. Σε κάθε γραμμή του πίνακα, όπου το υπόδειγμα λάθους εμφανίζεται (γραμμές 3 και 6), ο ΗΑΜΠ πίνακας σωστά θα συμπεράνει το ποια λέξη στάλθηκε. Επίσης, τουλάχιστον σε μια γραμμή (γραμμή 4), όπου εμφανίζεται το υπόδειγμα λάθους 110, αν 111 έχει σταλεί και 011 ληφθεί, ο ΗΑΜΠ λανθασμένα θα συμπεράνει ότι έχει σταλεί η 000. Παρατηρείστε ότι αυτός ο κώδικας διορθώνει τα υποδείγματα λάθους 000, 100, 010, και 001 όπου λαμβάνουν έναν αστερίσκο κάθε φορά που εμφανίζονται.

Παράδειγμα 1.12.3 Έστω $C = \{0000, 1010, 0111\}$. Ο ΗΑΜΠ πίνακας που αντιστοιχεί στον C είναι ο 1.2. Ο κώδικας C δε θα διορθώσει το υπόδειγμα λάθους $u = 1010$. Αυτό το υπόδειγμα λάθους εμφανίζεται στις γραμμές των $w = 0000$, 1010 και 1101. Στη μόνη περίπτωση που η ΗΑΜΠ σωστά συμπεραίνει ποια λέξη v έχει σταλεί, είναι στη $w = 1101$. Σημειώστε ότι το υπόδειγμα λάθους $u = 1010$ λαμβάνει έναν αστερίσκο μόνο στη στήλη για $v = 0111$ και σε καμιά άλλη από τις δύο στήλες. Ο C δε διορθώνει τα υποδείγματα λάθους 0000, 0100 και 0001.

Παράδειγμα 1.12.4 Έστω $C = \{001, 101, 110\}$. Διορθώνει ο C το υπόδειγμα λάθους $u = 100$; Κατασκευάζουμε μόνο τρεις γραμμές του ΗΑΜΠ πίνακα, όπου το 100 εμφανίζεται. Εφόσον $u = v + w$ και γνωρίζοντας τα u και v, μπορούμε να βρούμε τις ληφθείσες λέξεις από το $w = u + v$. Παρατηρείστε ότι το $u = 100$ δεν έχει αστερίσκο σε κάθε στήλη του ακόλουθου πίνακα, έτσι ο C δε διορθώνει το 100.

Ληφθείσα	Υπόδειγμα Λάθους			Αποκωδικοποίηση
w	$001 + w$	$101 + w$	$110 + w$	v
101	100	$000*$	011	101
001	$000*$	100	011	001
010	011	111	$100*$	110

Ασκήσεις

1.12.5 Έστω $C = \{001, 101, 110\}$. Ο C διορθώνει το υπόδειγμα λάθους $u = 100$; Το $u = 000$;

1.12.6 Αποδείξτε ότι το ίδιο υπόδειγμα λάθους δεν μπορεί να εμφανιστεί σε δοθείσα γραμμή ενός ΗΑΜΠ πίνακα.

1.12.7 Αποδείξτε ότι το υπόδειγμα λάθους 000, πάντα διορθώνεται.

1.12.8 Ποιό υπόδειγμα λάθους ο $C = K^n$ θα διορθώσει;

Η απόσταση ενός κώδικα, μπορεί να χρησιμοποιηθεί για την κατασκευή ενός τεστ διόρθωσης λαθών, το οποίο αποφεύγει τουλάχιστον κάποιο παραπάνω κόπο που συνεπάγεται ο έλεγχος του ΑΜΠ πίνακα. Το επόμενο θεώρημα μας δίνει αυτό το τεστ. Ας θυμηθούμε ότι το σύμβολο $\lfloor x \rfloor$, παριστάνει το μεγαλύτερο ακέραιο, μικρότερο ή ίσο του πραγματικού αριθμού x. Για παράδειγμα, $\lfloor 5/2 \rfloor = 2$, $\lfloor 3 \rfloor = 3$ και $\lfloor 1/2 \rfloor = 0$.

Θεώρημα 1.12.9 *Ένας κώδικας με απόσταση d θα διορθώνει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο με $\lfloor (d-1)/2 \rfloor$. Επίσης υπάρχει τουλάχιστον ένα υπόδειγμα λάθους με βάρος $1 + \lfloor (d-1)/2 \rfloor$ που ο C δε θα διορθώσει.*

Απόδειξη: Έστω u ένα υπόδειγμα λάθους με βάρος $wt(u) \leq (d-1)/2$. Έστω v και w δύο κωδικολέξεις του C, με $w \neq v$. Θέλουμε να αποδείξουμε ότι $d(v, v+u) < d(w, v+u)$.

$$\begin{aligned} d(w, v+u) + d(v, u+v) &\geq d(w, v) \\ &\geq d \\ d(w, v+u) + wt(u) &\geq 2wt(u) + 1 \\ d(w, v+u) &\geq wt(u) + 1 \\ &\geq d(v, v+u) + 1 \end{aligned}$$

αφού $wt(u) = d(v, v+u)$ και $2wt(u) + 1 \leq d$.

Ο C διορθώνει το u. Έστω τώρα v και w δύο κωδικολέξεις με $d(v,w) = d$. Σχηματίζουμε το υπόδειγμα λάθους u αλλάζοντας $d-1-\lfloor (d-1)/2 \rfloor$ από τους d άσους του $v+w$ σε μηδέν. Άρα

$$\begin{aligned} d(v, v+u) &= wt(u) = 1 + \lfloor (d-1)/2 \rfloor \text{ και} \\ d(w, v+u) &= wt(w+v+u) = d(v+w, u) \\ &= d - (1 + \lfloor (d-1)/2 \rfloor). \end{aligned}$$

Αν d είναι περιττό, έστω $d = 2t+1$, τότε:

$$\begin{aligned} d(v, v+u) &= wt(u) = 1 + (2t)/2 = 1 + t \text{ και} \\ d(w, v+u) &= 2t + 1 - (1+t) = t, \end{aligned}$$

άρα $d(v, v+u) > d(w, v+u)$. Αν d είναι άρτιο, έστω $d = 2t$, τότε:

$$\begin{aligned} d(v, v+u) &= 1 + \lfloor t - 1/2 \rfloor = t \text{ και} \\ d(w, v+u) &= 2t - t = t. \end{aligned}$$

Σε κάθε περίπτωση, $d(v, v+u) > d(w, v+u)$, οπότε το $v+u$ δεν είναι πιο κοντά στη v παρά στη w. Έτσι ο C δε διορθώνει το υπόδειγμα λάθους u.

Σύμφωνα με το παραπάνω θεώρημα, είναι φανερό ότι ένας κώδικας με απόσταση d είναι $\lfloor (d-1)/2 \rfloor$ κώδικας διόρθωσης λάθους.

Παράδειγμα 1.12.10 Ο κώδικας $C = \{000, 111\}$ έχει απόσταση $d = 3$. Επειδή $\lfloor (d-1)/2 \rfloor = 1$, το θεώρημα 1.12.9 μας εξασφαλίζει ότι ο C διορθώνει όλα τα υποδείγματα λάθους με βάρος 0 ή 1. Όπως παρατηρήσαμε στο παράδειγμα 1.12.1, ο C διορθώνει τα υποδείγματα λάθους $000, 100, 010$ και 001. Το υπόδειγμα λάθους 110 έχει βάρος $1 + \lfloor (d-1)/2 \rfloor = 2$ και είδαμε ότι ο C δεν το διορθώνει.

Το θεώρημα 1.12.9 δεν περιορίζει έναν κώδικα C με απόσταση d να μην μπορεί να διορθώσει υπόδειγμα λάθους με βάρος μεγαλύτερο του $\lfloor (d-1)/2 \rfloor$.

Παράδειγμα 1.12.11 Έστω $C = \{001, 101\}$. Τότε $d = 1$. Το υπόδειγμα λάθους $u = 011$, έχει βάρος 2, το οποίο είναι μεγαλύτερο του $1 + \lfloor (d-1)/2 \rfloor = 1$, όπως δείχνει το παρακάτω μέρος του ΑΜΠ πίνακα και έτσι, ο C διορθώνει το $u = 011$.

w	$001 + w$	$101 + w$	v
010	$011*$	111	001
110	111	$011*$	101

Ασκήσεις

1.12.12 Για καθέναν από τους παρακάτω κώδικες C:

(i) Προσδιορίστε τα υποδείγματα λάθους που ο C θα διορθώσει (οι ΗΑΜΠ πίνακες για αυτούς τους κώδικες, έχουν κατασκευαστεί στην άσκηση 1.9.7).

(ii) Βρείτε τα υποδείγματα λάθους που το θεώρημα 1.12.9 μας εξασφαλίζει ότι ο C θα διορθώσει.

(α'). $C = \{101, 111, 011\}$

(β'). $C = \{000, 001, 010, 011\}$

(γ'). $C = \{0000, 0001, 1110\}$

(δ'). $C = \{0000, 1001, 0110, 1111\}$

(ε'). $C = \{00000, 11111\}$

(ϛ'). $C = \{00000, 11100, 00111, 11011\}$

(ζ'). $C = \{00000, 11110, 01111, 10001\}$

(η'). $C = \{000000, 101010, 010101, 111111\}$

1.12.13 Χρησιμοποιώντας την τεχνική του Παραδείγματος 1.12.6, αποφασίστε ποια από τα παρακάτω υποδείγματα λάθους, διορθώνονται από τον κώδικα που τα συνοδεύει.

(α'). $C = \{000000, 100101, 010110, 001111, 110011, 101010, 011001, 111100\}$

(i) $u = 001000$

(ii) $u = 000010$

(iii) $u = 100100$

(β'). $C = \{1001011, 0110101, 1110010, 1111111\}$

(i) $u = 0100000$

(ii) $u = 0101000$

(iii) $u = 1100000$

1.12.14 Για κάθε κώδικα στην άσκηση 1.12.12, βρείτε ένα υπόδειγμα λάθους, βάρους $\lfloor (d-1)/2 \rfloor + 1$, που ο C δε διορθώνει.

1.12.15 Έστω C ο κώδικας που αποτελείται από όλες τις λέξεις μήκους 4 με άρτιο βάρος. Ορίστε τα υποδείγματα λάθους που ο C θα διορθώσει.

1.12.16 Έστω u_1 και u_2 υποδείγματα λάθους μήκους n και υποθέστε ότι τα u_1 και u_2 συμφωνούν τουλάχιστον στις θέσεις που εμφανίζεται κάποιος άσος στο u_1. Δείξτε ότι, αν ένας κώδικας C διορθώνει το u_2, τότε θα διορθώνει και το u_1.

Όπως έχουμε παρατηρήσει, τα υποδείγματα λάθους μικρού βάρους, είναι πιο πιθανό να συμβούν, παρά εκείνα με μεγάλο βάρος (θεώρημα 1.6.3). Οπότε, σχεδιάζοντας κάποιο κώδικα, θα πρέπει να συγκεντρώσουμε την προσοχή μας να έχει την ικανότητα να διορθώσει, ή τουλάχιστον να ανιχνεύσει υποδείγματα λάθους με μικρό βάρος.

Κεφάλαιο 2

Γραμμικοί κώδικες

2.1 Γραμμικοί κώδικες

Σ' αυτήν την ενότητα θα εισάγουμε μία ευρεία κλάση κωδίκων. Πράγματι, σχεδόν όλοι οι κώδικες που θα θεωρήσουμε ανήκουν σ' αυτήν την κλάση. Θα χρησιμοποιήσουμε κάποια ισχυρά μαθηματικά εργαλεία, τα οποία θα μας δώσουν τη δυνατότητα να λύσουμε κάποια από τα προηγούμενα προβλήματα της θεωρίας κωδίκων, όταν αυτά τα εργαλεία εφαρμοστούν σε κώδικες αυτής εδώ της κλάσης.

Ένας κώδικας C ονομάζεται *γραμικός κώδικας* αν $v + w$ είναι μία λέξη του C, όταν v και w ανήκουν στον C. Δηλαδή, ένας γραμμικός κώδικας, είναι ένας κώδικας ο οποίος είναι κλειστός ως προς την πρόσθεση των λέξεων. Για παράδειγμα ο $C = \{000, 111\}$ είναι ένας γραμμικός κώδικας, επειδή όλα τα αθροίσματα

$$000 + 000 = 000 \qquad 111 + 000 = 111$$

$$000 + 111 = 000 \qquad 111 + 111 = 000$$

ανήκουν στον C. Όμως ο $C_1 = \{000, 001, 101\}$ δεν είναι γραμμικός κώδικας, διότι οι 001 και 101 είναι στο C_1 αλλά όχι η $001 + 101$.

Ένας γραμμικός κώδικας πρέπει να περιέχει τη μηδενική λέξη. Διότι αν ο C είναι πράγματι γραμμικός, τότε το άθροισμα $v + v = 0$ πρέπει να ανήκει στον C από την κλειστότητα ως προς την άθροιση. Εντούτοις όπως φαίνεται και από τον C_1, η μηδενική λέξη μέσα σε ένα κώδικα δε μας εξασφαλίζει ότι ο κώδικας είναι γραμμικός.

Ασκήσεις

2.1.1 Προσδιορίστε ποιοι από τους παρακάτω κώδικες είναι γραμμικοί.

(α'). $C = \{101, 111, 011\}$

(β'). $C = \{000, 001, 010, 011\}$

(γ'). $C = \{0000, 0001, 1110\}$

(δ΄). $C = \{0000, 1001, 0110, 1111\}$

(ε΄). $C = \{00000, 11111\}$

(ϛ΄). $C = \{00000, 11100, 00111, 11011\}$

(ζ΄). $C = \{00000, 11110, 01111, 10001\}$

(η΄). $C = \{000000, 101010, 010101, 111111\}$

Ένα πλεονέκτημα ενός γραμμικού κώδικα ως προς ένα μη γραμμικό κώδικα, είναι ότι η απόσταση βρίσκεται ευκολότερα. *Η απόσταση ενός γραμμικού κώδικα είναι ίση με το ελάχιστο βάρος μιας μη μηδενικής κωδικολέξης.* Η άσκηση 2.1.4 παρακάτω μας ζητάει να το αποδείξουμε.

Ασκήσεις

2.1.2 Δείξτε ότι ο $C = \{0000, 1100, 0011, 1111\}$ είναι γραμμικός κώδικας και ότι έχει απόσταση $d = 2$.

2.1.3 Βρείτε την απόσταση κάθε γραμμικού κώδικα στην άσκηση 2.1.1. Ελέξτε τις απαντήσεις με την άσκηση 1.11.12.

2.1.4 Δείξτε ότι η απόσταση ενός γραμμικού κώδικα είναι το βάρος της μηδενικής κωδικολέξης με το ελάχιστο βάρος.

Όπως θα δούμε στις επόμενες ενότητες, οι γραμμικοί κώδικες είναι μάλλον αρκετά δομημένοι και έχουν πολλά πλεονεκτήματα σε σχέση με τους τυχαίους κώδικες που μέχρι τώρα έχουμε συζητήσει. Τα παρακάτω είναι κάποια προβλήματα, δύκολα να αντιμετωπιστούν γενικά, αλλά σχετικά εύκολα για γραμμικούς κώδικες:

1) Για γραμμικούς κώδικες υπάρχει ένας αλγόριθμος για την ΑΜΠ, ο οποίος είναι απλούστερος και γρηγορότερος να χρησιμοποιήσουμε, απ' ότι εκείνο που περιγράψαμε (πριν ακόμα κάποιοι ειδικοί γραμμικοί κώδικες με απλούστερη δομή έχουν πολύ απλούς αλγόριθμους αποκωδικοποίησης).

2) Η κωδικοποίηση ενός γραμμικού κώδικα είναι ταχύτερη και απαιτεί λιγότερο αποθηκευτικό χώρο απ' ότι ένας τυχαίος μη γραμμικός κώδικας.

3) Οι πιθανότητες $\theta_p(C, v)$ είναι εύκολες να υπολογιστούν για ένα γραμμικό κώδικα.

4) Είναι εύκολο να περιγράψουμε το σύνολο των υποδειγμάτων λάθους που ένας γραμμικός κώδικας θα ανιχνεύσει.

5) Είναι πολύ πιο εύκολο να περιγράψουμε τα υποδείγματα λάθους που ένας γραμμικός κώδικας θα διορθώσει, παρά σε έναν τυχαίο μη γραμμικό κώδικα.

Τα πιο σημαντικά εργαλεία και τεχνικές για την μελέτη των γραμμικών κωδικών έρχονται από τη γραμμική άλγεβρα. Σ' αυτήν αλλά και σε αρκετές από τις επόμενες ενότητες θα δούμε περιληπτικά κάποιες βασικές προτάσεις από τη γραμμική άλγεβρα και θα προσπαθήσουμε να δούμε τη σχέση τους με τη θεωρία κωδίκων. Οι περισσότερες από τις αποδείξεις που δεν εξαρτώνται από τα γινόμενα στοιχείων του K^n με αριθμούς είναι ακριβώς επανάληψη των αποείξεων στο $\mathbb{R}^n$ και γι' αυτό παραλείπονται.

Ας επαναλάβουμε εδώ ότι έχουμε ορίσει ένα διανυσματικό χώρο (υπέρ του K), που αποτελείται από αριθμούς (τα ψηφία 0 και 1 του K) και ένα σύνολο από διανύσματα ή λέξεις, το K^n και μαζί με τη διανυσματική πρόσθεση και τον πολλαπλασιασμό διανύσματος με αριθμό, ο οποίος ικανοποιεί όλες τις ιδιότητες που περιγράψαμε στην ενότητα 1.7. Ένα μη κενό υποσύνολο U ενός διανυσματικού χώρου V, λέγεται *υπόχωρος* του V, αν ο U είναι κλειστός ως προς τη διανυσματική πρόσθεση και το βαθμωτό πολλαπλασιασμό· δηλαδή, αν v και w είναι στοιχεία του U, τότε τα $v + w$ και av ανήκουν στο U για οποιοδήποτε αριθμό a. Ιδιαίτερα επειδή οι μόνοι αριθμοί στο K είναι το 0 και το 1, ο U είναι υπόχωρος του K^n αν και μόνο αν ο U είναι κλειστός ως προς την πρόσθεση. Έτσι ο C είναι γραμμικός κώδικας αν και μόνο αν ο C είναι ένας υπόχωρος του K^n. Στις επόμενες ενότητες θα χρησιμοποιήσουμε τις γνώσεις μας ώστε να βελτιώσουμε δραματικά τις μεθόδους της κωδικοποίησης και της αποκωδικοποίησης.

2.2 Δύο σημαντικοί υπόχωροι

Θα θεωρήσουμε δύο υπόχωρους του διανυσματικού χώρου K^n οι οποίοι θα μας δώσουν δύο ενδιαφέροντα παραδείγματα γραμμικών κωδίκων και θα είναι ζωτικοί σε μελλοντικές μελέτες. Οι ορισμοί και τα αποτελέσματα θα δοθούν για ένα τυχαίο διανυσματικό χώρο και μετά για τον K^n.

Ένα διάνυσμα w είναι ένας *γραμμικός συνδυασμός* διανυσμάτων $v_1, v_2, ..., v_k$ αν υπάρχουν αριθμοί $a_1, a_2, ..., a_k$ έτσι ώστε

$$w = a_1v_1 + a_2v_2 + ... + a_kv_k.$$

Το σύνολο όλων των γραμμικών συνδυασμών των διανυσμάτων σε ένα δοσμένο σύνολο $S = \{v_1, v_2, ..., v_k\}$ ονομάζεται *γραμμικό ανάπτυγμα* του S και συμβολίζεται με $< S >$. Αν το S είναι κενό, τότε ορίζουμε $< S >= \{0\}$.

Στη γραμμική άλγεβρα αποδεικνύεται ότι για κάθε υποσύνολο S ενός διανυσματικού χώρου V, το γραμμικό ανάπτυγμα $< S >$ είναι ένας υπόχωρος του V, που ονομάζει τον υπόχωρο *αναπτυγμένο* ή *γεννημένο* από το S. Για το γραμμικό χώρο K^n, έχουμε απλή περιγραφή του $< S >$, η οποία αποτελεί το παρακάτω θεώρημα. Επειδή το $< S >$ είναι ένας υπόχωρος του K^n θα καλούμε το $< S >$ ως το γραμμικό κώδικα που γεννιέται από το S.

Θεώρημα 2.2.1 *Για κάθε υποσύνολο S του K^n, ο κώδικας $C =< S >$ που γεννιέται από το S αποτελείται ακριβώς από τις παρακάτω λέξεις: τη μηδενική λέξη, όλες τις λέξεις στο S και όλα τα αθροίσματα δύο ή περισσοτέρων λέξεων στο S.*

Παράδειγμα 2.2.2 Έστω $S = \{0100, 0011, 1100\}$. Τότε ο κώδικας $C =< S >$ που γεννιέται από το S αποτελείται από τις

$$\begin{array}{llll} 0000, & 0100, & 0100+0011=0111, & 0100+0011+1100=1011, \\ 1100, & 0011, & 0100+1100=1000, & 0011+1100=1111\cdot \end{array}$$

δηλαδή $C =< S >= \{0000, 0100, 0011, 1100, 0111, 1000, 1011, 1111\}$.

Ασκήσεις

2.2.3 Για κάθε ένα από τα παρακάτω σύνολα S, γράψτε όλα τα στοιχεία του γραμμικού κώδικα $< S >$.

(α΄). $C = \{010, 011, 111\}$

(β΄). $C = \{1010, 0101, 1111\}$

(γ΄). $C = \{0101, 1010, 1100\}$

(δ΄). $C = \{1000, 0100, 0010, 0001\}$

(ε΄). $C = \{11000, 01111, 11110, 01010\}$

(ϛ΄). $C = \{10101, 01010, 11111, 00011, 10110\}$

Αν $v = \{a_1, a_2, ..., a_n\}$ και $w = \{b_1, b_2, ..., b_n\}$ είναι διανύσματα του K^n, ορίζουμε το *εσωτερικό γινόμενο* $v \cdot w$ των v και w ως

$$v \cdot w = a_1b_1 + a_2b_2 + ... + a_nb_n.$$

Σημειώστε ότι $v \cdot w$ είναι ένας αριθμός. Για παράδειγμα, στο K^5,

$$\begin{aligned} 11001 \cdot 01101 &= 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 \\ &= 0 + 1 + 0 + 0 + 1 \\ &= 0. \end{aligned}$$

Ασκήσεις

2.2.4 Κατασκευάστε παραδείγματα στο K^5 για κάθε έναν από τους παρακάτω κανόνες

(α΄). $u \cdot (v + w) = u \cdot v + u \cdot w$

(β΄). $a(v \cdot w) = (av) \cdot w = v \cdot (aw)$.

2.2.5 Αποδείξτε ότι οι δύο κανόνες της άσκησης 2.2.4 ισχύουν στον K^n.

Δύο διανύσματα v και w είναι *ορθογώνια* αν $v \cdot w = 0$. Το παραπάνω παράδειγμα δείχνει ότι τα $v = 11001$ και $w = 01101$ είναι ορθογώνια στο K^5. Για ένα δοσμένο σύνολο S από διανύσματα του K^n, θα λέμε ότι το διάνυσμα v είναι *ορθογώνιο στο σύνολο* S αν $v \cdot w = 0$ για όλα τα w στο $S^{\cdot}$ δηλαδή το v είναι ορθογώνιο σε κάθε διάνυσμα του S. Το σύνολο όλων των διανυσμάτων που είναι ορθογώνια στο S συμβολίζεται με $S^{\perp}$ και ονομάζεται *ορθογώνιο συμπλήρωμα* του S.

Στη Γραμμική Άλγεβρα αποδεικνύεται ότι για κάθε υποσύνολο του S ενός διανυσματικού χώρου V, το ορθογώνιο συμπλήρωμα $S^{\perp}$ είναι ένας υπόχωρος του V. Για το διανυσματικό χώρο K^n, αν $C =< S >$, τότε θα γράφουμε $C^{\perp} = S^{\perp}$ και θα ονομάζουμε το $C^{\perp}$ *δυϊκό κώδικα* του C.

Παράδειγμα 2.2.6 Για $S = \{0100, 0101\}$, υπολογίζουμε το δυϊκό κώδικα $C^{\perp} = S^{\perp}$. Πρέπει να βρούμε όλες τις λέξεις $v = (x, y, z, w)$ στο K^4 έτσι ώστε και οι δύο εξισώσεις

$$v \cdot 0100 = 0$$
$$v \cdot 0101 = 0$$

να ικανοποιούνται. Υπολογίζοντας το εσωτερικό γινόμενο έχουμε

$$y = 0 \text{ και } y + w = 0.$$

Οπότε $y = w = 0$ αλλά τα x και z μπορεί να είναι 0 ή 1. Γράφοντας όλες τις επιλογές για τη v, παίρνουμε

$$C^{\perp} = S^{\perp} = \{0000, 0010, 1000, 1010\}.$$

Ασκήσεις

2.2.7 Βρείτε το δυϊκό κώδικα $C^{\perp}$ για κάθε έναν από τους κώδικες $C =< S >$ της άσκησης 2.2.3.

2.2.8 Βρείτε ένα παράδειγμα μιας μη μηδενικής λέξης v τέτοιας ώστε $v \cdot v = 0$. Τι μπορείτε να πείτε για το βάρος μιας τέτοιας λέξης;

2.2.9 Για κάθε υποσύνολο S ενός διανυσματικού χώρου V, ισχύει $(S^{\perp})^{\perp} =< S >$. Χρησιμοποιήστε το παράδειγμα 2.2.6 για να κατασκευάσετε ένα παράδειγμα της παραπάνω πρότασης.

2.2.10 Δείξτε ότι $< S > \subseteq (S^{\perp})^{\perp}$. (Οπότε $(S^{\perp})^{\perp} =< S >^{\cdot}$ για ένα γραμμικό κώδικα C, αυτό σημαίνει $(C^{\perp})^{\perp} = C$).

2.3 Ανεξαρτησία, Βάση, Διάσταση

Θα επαναλάβουμε περιληπτικά κάποιες σπουδαίες έννοιες από τη Γραμμική Άλγεβρα και θα δείξουμε πως θα τις εφαρμόσουμε σε γραμμικούς κώδικες. Ο

κύριος σκοπός είναι να βρούμε έναν αποτελεσματικό τρόπο να περιγράψουμε ένα γραμμικό κώδικα χωρίς να πρέπει να γράψουμε όλες τις κωδικολέξεις του.

Ένα σύνολο $S = \{v_1, v_2, ..., v_k\}$ από διανύσματα είναι *γραμμικά εξαρτημένο* αν υπάρχουν αριθμοί $a_1, a_2, ..., a_k$ όχι όλοι μηδέν έτσι ώστε

$$a_1v_1 + a_2v_2 + ... + a_kv_k = 0.$$

Αλλιώς το σύνολο S είναι *γραμμικά ανεξάρτητο*.

Ο έλεγχος λοιπόν για γραμμική ανεξαρτησία είναι να σχηματίσουμε την παραπάνω διανυσματική εξίσωση χρησιμοποιώντας τυχαίους αριθμούς. Αν αυτή η εξίσωση εξαναγκάσει *όλα* τα νούμερα $a_1, a_2, ..., a_k$ να είναι 0, τότε το S είναι γραμμικά ανεξάρτητο. Αν *τουλάχιστον ένα* a_1 μπορεί να επιλεγεί να είναι μη μηδενικό τότε το S είναι γραμμικά εξαρτημένο.

Παράδειγμα 2.3.1 Ελέγχουμε το $S = \{1001, 1101, 1011\}$ για γραμμική ανεξαρτησία. Έστω a, b και c είναι αριθμοί (ψηφία) έτσι ώστε

$$a(1001) + b(1101) + c(1011) = 0000.$$

Εξισώνοντας τις δύο πλευρές παίρνουμε τις εξισώσεις

$$a + b + c = 0, b = 0, c = 0, a + b + c = 0.$$

Αυτές οι εξισώσεις εξαναγκάζουν $a = b = c = 0$. Οπότε το S είναι ένα γραμμικά ανεξάρτητο σύνολο λέξεων στο K^4.

Παράδειγμα 2.3.2 Ελέγχουμε το $S = \{110, 011, 101, 111\}$ για γραμμική ανεξαρτησία. Θεωρούμε

$$a(110) + b(011) + c(101) + d(111) = 000.$$

Αυτό παράγει το σύστημα εξισώσεων

$$\begin{aligned} a + c + d &= 0 \\ a + b + d &= 0 \\ b + c + d &= 0. \end{aligned}$$

Προσθέτοντας τις τρεις εξισώσεις παίρνουμε $d = 0$. Τώρα έχουμε $a + c = 0$, $a + b = 0$, $b + c = 0$. Έτσι μπορούμε να επιλέξουμε $a = b = c = 1$. Οπότε το S είναι ένα γραμμικά εξαρτημένο σύνολο.

Στη Γραμμική Άλγεβρα αποδεικνύεται ότι *κάθε σύνολο από διανύσματα* $S \neq \{0\}$ *περιέχει ένα μέγιστο γραμμικά ανεξάρτητο υποσύνολο*. Το παρακάτω παράδειγμα δείχνει πως ένα τέτοιο υποσύνολο μπορεί να βρεθεί.

Παράδειγμα 2.3.3 Έστω $S = \{110, 011, 101, 111\}$. Το τελευταίο παράδειγμα δείχνει ότι το S είναι γραμμικά εξαρτημένο. Πράγματι, βρίσκουμε ότι

$$1(110) + 1(011) + 1(101) + 1(111) = 000,$$

οπότε μπορούμε να λύσουμε ως προς 101 ως γραμμικός συνδυασμός των άλλων λέξεων στο S:

$$101 = 1(110) + 1(011) + 0(111).$$

Στο εξαρτημένο σύνολο S, αν πάρουμε τις λέξεις με τη σειρά που δίνονται, καταλήγουμε στην 101 να είναι η πρώτη εξαρτημένη λέξη, δηλαδή, είναι ένας γραμμικός συνδυασμός των δύο προηγούμενων λέξεων 110 και 011 του S. Αφαιρώντας αυτήν τη λέξη φτάνουμε στο νέο σύνολο $S' = \{110, 011, 111\}$. Τώρα μπορούμε να ελέγξουμε το S' για γραμμική ανεξαρτησία. Αν το S' είναι γραμμικά εξαρτημένο, τότε αφαιρούμε την πρώτη λέξη που είναι γραμμικός συνδυασμός των προηγούμενων, έτσι φτιάχνουμε ένα νέο σύνολο S''. Αυτή η διαδικασία μπορεί να επαναληθφεί έως ότου βρούμε ένα νέο σύνολο το οποίο είναι γραμμικά ανεξάρτητο· τέτοιο σύνολο είναι πάντοτε το μεγαλύτερο γραμμικά ανεξάρτητο υποσύνολο του δοθέντος συνόλου S. Στο παρόν παράδειγμα αυτό το σύνολο είναι το S'.

Ασκήσεις

2.3.4 Ελέγξτε κάθε ένα από τα παρακάτω σύνολα για γραμμική ανεξαρτησία. Αν το σύνολο είναι γραμμικά εξαρτημένο, ξεχωρίστε από το S ένα μέγιστο γραμμικά ανεξάρτητο υποσύνολο.

(α'). $S = \{1101, 1110, 1011\}$

(β'). $C = \{101, 011, 110, 010\}$

(γ'). $S = \{1101, 0111, 1100, 0011\}$

(δ'). $C = \{1000, 0100, 0010, 0001\}$

(ε'). $S = \{1000, 1100, 1110, 1111\}$

(ϛ'). $S = \{1100, 1010, 1001, 0101\}$

(ζ'). $C = \{0110, 1010, 1100, 0011, 1111\}$

(η'). $S = \{111000, 000111, 101010, 010101\}$

(θ'). $S = \{00000000, 10101010, 01010101, 11111111\}$

Στην άσκηση 2.3.4 (9.) το S θα βρεθεί να είναι ένα γραμμικό εξαρτημένο σύνολο. Παρατηρείστε ότι το S περιέχει τη μηδενική λέξη. Είναι πάντοτε αλήθεια ότι *κάθε σύνολο διανυσμάτων που περιέχει το μηδενικό διάνυσμα είναι γραμμικά εξαρτημένο.*

Ένα μη κενό υποσύνολο B διανυσμάτων ενός γραμμικού χώρου V είναι μία *βάση* του V αν ισχύουν τα δύο παρακάτω:

1) Το B παράγει το V (δηλαδή $< B >= V$) και

2) Το B είναι ένα γραμμικά ανεξάρτητο σύνολο.

Σημειώστε ότι ένα *οποιοδήποτε γραμμικά ανεξάρτητο σύνολο B είναι αυτόματα μία βάση για το $< B >$* . Ακόμα επειδή κάθε γραμμικά εξαρτημένο σύνολο S διανυσμάτων περιέχει μία μη μηδενική λέξη, θα περιέχει και ένα μέγιστο γραμμικά ανεξάρτητο υποσύνολο B, οπότε μπορούμε να ξεχωρίσουμε από το S μία βάση B για το $< S >$. Εάν $S = \{0\}$ τότε λέμε ότι η βάση του S είναι το κενό σύνολο, $\emptyset$.

Παράδειγμα 2.3.5 Έστω $S = \{1001, 1101, 1011\}$. Στο παράδειγμα 2.3.1 βρήκαμε ότι το S είναι γραμμικά ανεξάρτητο. Οπότε το S είναι μία βάση για τον κώδικα $C =< S >= \{0000, 1001, 1101, 1011, 0100, 0010, 0110, 1111\}$ ο οποίος είναι ένας υπόχωρος του K^4.

Παράδειγμα 2.3.6 Έστω $S = \{110, 011, 101, 111\}$. Στο παράδειγμα 2.3.2 βρήκαμε ότι το S είναι γραμμικά εξαρτημένο. Αλλά στο παράδειγμα 2.3.3 ξεχωρίσαμε ένα μέγιστο γραμμικά ανεξάρτητο υποσύνολο $B = S' = \{110, 011, 111\}$ του S. Οπότε το B είναι μία βάση για τον κώδικα $C =< S >$.

Αυτά τα παραδείγματα μας δείχνουν πως θα παράγουμε μια βάση για τον κώδικα $C =< S >$ που γεννιέται από ένα μη κενό υποσύνολο S του K^n. Για να βρούμε μία βάση του δυϊκού κώδικα $C^\perp$, ξεχωρίζουμε ένα μέγιστο γραμμικά ανεξάρτητο υποσύνολο του $C^\perp$ ακολουθώντας τη διαδικασία του παραδείγματος 2.3.3.

Ασκήσεις

2.3.7 Για κάθε ένα υποσύνολο της άσκησης 2.3.3 βρείτε μία βάση B για τον κώδικα $C =< S >$ και μια βάση $B^\perp$ του δυϊκού κώδικα $C^\perp$.

Το σύνολο $B = \{110, 011, 111\}$ δεν είναι το μοναδικό μέγιστο γραμμικά ανεξάρτητο υποσύνολο του $S = \{110, 011, 101, 111\}$ (δείτε παράδειγμα 2.3.6). Το σύνολο $B_1 = \{110, 101, 111\}$ είναι επίσης ένα τέτοιο υποσύνολο του S. Έτσι το B_1 είναι επίσης μία βάση για τον κώδικα $C =< S >$.

Γενικά ένας γραμμικός χώρος συνήθως έχει πολλές βάσεις. Όμως, *όλες οι βάσεις για ένα διανυσματικό χώρο περιέχουν το ίδιο πλήθος στοιχείων.* Το πλήθος των στοιχείων μιας οποιασδήποτε βάσης ενός διανυσματικού χώρου καλείται *διάσταση* του χώρου.

Η διάσταση του K^n είναι n, διότι το σύνολο όλων των λέξεων μήκους n και βάρους ένα, αποτελεί μία βάση του K^n. Αντίθετα, η βάση του υποχώρου $\{0\}$ είναι το $\emptyset$ έτσι έχει διάσταση 0.

Ασκήσεις

2.3.8 Βρείτε τις διαστάσεις για κάθε έναν από τους κώδικες $C =< S >$ και του δυϊκού του $C^\perp$ στην άσκηση 2.2.3 (δείτε επίσης και την άσκηση 2.2.7).

Μία βάση μας δίνει έναν αποτελεσματικό τρόπο να περιγράψουμε ένα γραμμικό κώδικα. Για κάθε διανυσματικό χώρο V, αν η $\{v_1, v_2, ..., v_k\}$ *είναι μία βάση για τον V, τότε κάθε διάνυσμα w στο V μπορεί να γραφτεί ως ένας μοναδικός γραμμικός συνδυασμός των βασικών διανυσμάτων $v_1, v_2, ..., v_k$*· δηλαδή υπάρχουν μοναδικοί αριθμοί $\{a_1, a_2, ..., a_k\}$ έτσι ώστε $w = a_1v_1 + a_2v_2 + ... + a_kv_k$.

Παράδειγμα 2.3.9 Θα γράψουμε τη $w = 011$ ως ένα μοναδικό γραμμικό συνδυασμό λέξεων της βάσης $\{110, 001, 100\}$ του K^3. Ζητάμε ψηφία a, b, c έτσι ώστε

$$a(110) + b(001) + c(100) = 011.$$

Αυτό παράγει τις αριθμητικές εξισώσεις

$$a + c = 0, a = 1, b = 1,$$

οι οποίες έχουν μοναδική λύση $a = b = c = 1$. Οπότε $011 = 1(110) + 1(001) + 1(100)$.

Ασκήσεις

2.3.10 Να γράψετε κάθε μία από τις παρακάτω λέξεις του K^4 ως ένα μοναδικό γραμμικό συνδυασμό των λέξεων της βάσης $\{1000, 1100, 1110, 1111\}$:

(α'). 0011

(β'). 1010

(γ'). 0111

(δ'). 0001

(ε'). 0000.

Άλλο σπουδαίο γεγονός για τους διανυσματικούς χώρους είναι ότι *κάθε γραμμικά ανεξάρτητο υποσύνολο ενός διανυσματικού χώρου περιέχεται σε μία βάση του χώρου*. Το επόμενο παράδειγμα αποδεικνύει πως πετυχαίνεται αυτό.

Παράδειγμα 2.3.11 Το σύνολο $S = \{110, 001\}$ είναι ένα γραμμικά ανεξάρτητο σύνολο του K^3. Επεκτείνουμε το S σε μία βάση για το K^3. Κατ'αρχάς προσαρτούμε στο S μια οποιαδήποτε γνωστή βάση: 100, 010, 001 είναι μία βολική βάση του K^3 για να την προσαρτήσουμε. Τη λίστα των λέξεων που προκύπτει

$$110, 001, 100, 010, 001$$

τη συρικνώνουμε σε μία βάση του K^3 με βάση τη διαδικασία του παραδείγματος 2.3.3, λύνοντας ως προς 100, 010 ή 001.

Ασκήσεις

2.3.12 (α'). Βρείτε μία βάση του K^4 που περιέχει το $\{1001, 1101\}$.

(β'). Επεκτείνετε το $\{101010, 010101\}$ σε μία βάση του K^6.

Τώρα ερχόμαστε σε δύο σημαντικά θεωρήματα που αφορούν τη διάσταση ενός γραμμικού κώδικα. Αν ένας γραμμικός κώδικας C έχει διάσταση k και αν

$\{v_1, v_2, ..., v_k\}$ είναι μία βάση για τον C, τότε μία λέξη w του C μπορεί να γραφτεί ως

$$w = a_1v_1 + a_2v_2 + ... + a_kv_k$$

για μία μοναδική επιλογή των ψηφίων $a_1, a_2, ..., a_k$. Αφού a_1 είναι 1 ή 0, υπάρχουν 2^k επιλογές για τα $a_1, a_2, ..., a_k$ και άρα 2^k λέξεις στο C.

Θεώρημα 2.3.13 *Ένας γραμμικός κώδικας διάστασης k περιέχει ακριβώς 2^k κωδικολέξεις.*

Το παρακάτω θεώρημα μπορεί να αποδειχτεί χρησιμοποιώντας στοιχειώδη αποτελέσματα από τη θεωρία των συστημάτων των γραμμικών εξισώσεων.

Θεώρημα 2.3.14 *Έστω $C =< S >$ είναι ο γραμμικός κώδικας που γεννιέται από ένα υποσύνολο S του K^n. Τότε ισχύει (διάσταση του C) + (διάσταση του $C^{\perp}$) $= n$.*

Ασκήσεις

2.3.15 Ελέγξτε τις απαντήσεις σας στην άσκηση 2.3.8 με την εξίσωση του θεωρήματος 2.3.14.

2.3.16 Έστω S είναι ένα υποσύνολο του K^7, έστω $C =< S >$ και υποθέτουμε ότι ο $C^{\perp}$ έχει διάσταση 3.

(α'). Βρείτε τη διάσταση του $C =< S >$.

(β'). Βρείτε το πλήθος των λέξεων του C.

2.3.17 Έστω S είναι ένα υποσύνολο του K^8 και υποθέτουμε ότι το $\{11110000, 00001111, 10000001\}$ είναι μία βάση του $C^{\perp}$. Βρείτε το πλήθος των λέξεων του $C =< S >$.

2.3.18 Το θεώρημα 2.3.14 ισχύει και στο $\mathbb{R}^n$. Στο $\mathbb{R}^n$ κάθε διάνυσμα μπορεί να γραφεί μοναδικά ως το άθροισμα ενός διανύσματος του $< S >$ και ενός του $S^{\perp}$ και το μηδενικό διάνυσμα είναι το μοναδικό διάνυσμα που είναι κοινό και στον $< S >$ και στον $S^{\perp}$. (Για παράδειγμα, στο $\mathbb{R}^3$ παίρνουμε ως $< S >$ το xy επίπεδο και ως $S^{\perp}$ τον άξονα z). Χρησιμοποιείστε το $S = \{000, 101\}$ στο K^3 για να δείξετε ότι αυτό δεν ισχύει γενικά στο K^n.

Το τελευταίο αποτέλεσμα σ' αυτήν την ενότητα ασχολείται με το ερώτημα πόσες δοαφορετικές βάσεις έχει ένας γραμμικός κώδικας. Στο $\mathbb{R}^n$ ένας υπόχωρος μπορεί να έχει άπειρες βάσεις, αλλά δε συμβαίνει το ίδιο στο K^n.

Θεώρημα 2.3.19 *Ένας γραμμικός κώδικας με διάσταση k έχει ακριβώς $\frac{1}{k!}\prod_{i=0}^{k-1}(2^k - 2^i)$ διαφορετικές βάσεις.*

Παράδειγμα 2.3.20 Ο γραμμικός κώδικας K^4 έχει διάσταση $k = 4$ και έτσι

$$\frac{1}{4!}\prod_{i=0}^{3}(2^4 - 2^i) = \frac{1}{4!}(2^4 - 1)(2^4 - 2)(2^4 - 2^2)(2^4 - 2^3) = 840$$

διαφορετικές βάσεις. Κάθε γραμμικός κώδικας που περιέχεται στο K^n, για $n \geq 4$, ο οποίος έχει διάσταση 4 έχει επίσης 840 διαφορετικές βάσεις.

Ασκήσεις

2.3.21 Έστω b_n είναι το πλήθος των διαφορετικών βάσεων του K^n. Ελέγξτε τα νούμερα στον παρακάτω πίνακα:

n	1	2	3	4	5	6
b_n	1	3	28	840	83.328	27.998.208

2.3.22 Καταγράψτε όλες τις βάσεις του K^2 και του K^3.

2.3.23 Βρείτε το πλήθος των διαφορετικών βάσεων για κάθε ένα κώδικα $C = < S >$ όπου

(α'). $C = \{010, 011, 111\}$

(β'). $C = \{1010, 0101, 1111\}$

(γ'). $C = \{0101, 1010, 1100\}$

(δ'). $C = \{1000, 0100, 0010, 0001\}$

(ε'). $C = \{11000, 01111, 11110, 01010\}$

(ϛ'). $C = \{10101, 01010, 11111, 00011, 10110\}$

2.4 Πίνακες

Ένας $m \times n$ πίνακας είναι ένα παραλληλόγραμμο με αριθμούς που έχουν διαταχθεί σε m γραμμές και n στήλες. Υποθέτουμε ότι ο αναγνώστης είναι γνώστης της άλγεβρας των πινάκων επί των πραγματικών αριθμών. Σ' αυτήν την ενότητα υπενθυμίζουμε τα αναγκαία εργαλεία της στοιχειώδους θεωρίας πινάκων που χρειαζόμαστε για τη θεωρία κωδίκων.

Αν A είναι ένας $m \times n$ πίνακας και B ένας $n \times p$ πίνακας, τότε το γινόμενο $A \cdot B$ είναι ο $m \times p$ πίνακας, ο οποίος έχει στη θέση (i, j), (δηλαδή στη θέση στη γραμμή i και στη στήλη j), το εσωτερικό γινόμενο της i γραμμής του A και της j στήλης του B. Για παράδειγμα

$$\begin{bmatrix} 1011 \\ 0101 \end{bmatrix} \cdot \begin{bmatrix} 101 \\ 011 \\ 101 \\ 100 \end{bmatrix} = \begin{bmatrix} 100 \\ 111 \end{bmatrix}.$$

Σημειώστε ότι το πλήθος των στηλών του πρώτου πίνακα πρέπει να είναι ίσο με το πλήθος των γραμμών του δεύτερου πίνακα έτσι ώστε το γινόμενο να ορίζεται.

Ασκήσεις

2.4.1 Βρείτε το γινόμενο κάθε ζευγαριού από τους παρακάτω πίνακες σε κάθε περίπτωση που ορίζεται το γινόμενο.

$$A = \begin{bmatrix} 11011 \\ 00101 \\ 11011 \end{bmatrix}, B = \begin{bmatrix} 0101 \\ 1001 \\ 1100 \end{bmatrix}, C = \begin{bmatrix} 110110 \\ 011011 \\ 101011 \end{bmatrix}, D = \begin{bmatrix} 1111 \\ 0101 \\ 1010 \\ 1101 \end{bmatrix}.$$

Οι συνήθεις αλγεβρικοί κανόνες για πίνακες πάνω απόεπί των πραγματικών ισχύουν και για πίνακες επί του K. Ο $m \times n$ *μηδενικός πίνακας* είναι ο $m \times n$ πίνακας με κάθε θέση του ίση με το 0. Ο $n \times n$ (τετραγωνικός) πίνακας I στον οποίο η (i, j) θέση του είναι 1 αν $i = j$, αλλιώς είναι 0 είναι ο $n \times n$ *ταυτοτικός πίνακας*. Για κάθε πίνακα A, ισχύουν $AI = A$ και $IA = A$. Οι παρακάτω τρεις ασκήσεις τονίζουν τρεις αλγεβρικούς νόμους που δεν ισχύουν για πίνακες υπέρ του K.

Ασκήσεις

2.4.2 Βρείτε δύο 2×2 πίνακες επί του K έτσι ώστε $AB \neq BA$.

2.4.3 Βρείτε 2×2 πίνακες A και B επί του K και οι δύο διαφορετικοί από το μηδενικό πίνακα, έτσι ώστε $A \cdot B = 0$.

2.4.4 Βρείτε 2×2 πίνακες A, B και C υπέρ του K τέτοιοι ώστε $AB = AC$ και $B \neq C$.

Υπάρχουν δύο τύποι *στοιχειωδών πράξεων σε γραμμές* που μπορούν να εφαρμοστούν σε έναν πίνακα υπέρ του K. Αυτοί είναι:

(α'). Εναλλαγή δύο γραμμών και

(β'). Αντικατάσταση μιας γραμμής με το άθροισμα του εαυτού της και μιας οποιασδήποτε άλλης γραμμής.

Δύο πίνακες λέγονται *ισοδύναμοι ως προς τις γραμμές (ή γραμμικά ισοδύναμοι)* αν ο ένας προέρχεται από τον άλλο με μια ακολουθία από στοιχειώδεις πράξεις σε γραμμές.

Μία μονάδα σε έναν πίνακα M (υπέρ του K) ονομάζεται *οδηγός μονάδα* αν δεν υπάρχουν μονάδες από τα αριστερά στην ίδια γραμμή και μία στήλη του M ονομάζεται *οδηγός στήλη* αν περιέχει μία τουλάχιστον οδηγό μονάδα. Ο M βρίσκεται σε *κλιμακωτή μορφή γραμμών - row echelon form* (ή γραμμοκλιμακωτή μορφή σε συντομία ΓΚΜ, από τα αρχικά των λέξεων) αν οι μηδενικές γραμμές του M (αν βέβαια υπάρχουν τέτοιες) είναι όλες στο κάτω μέρος του M και κάθε οδηγός μονάδα βρίσκεται στα δεξιά της οδηγού μονάδας της προηγούμενης γραμμής. Αν επιπλέον κάθε οδηγός στήλη περιέχει ακριβώς μία μονάδα, τότε ο M είναι σε ανηγμένη γραμμοκλιμακωτή μορφή (reduced row echelon form)σε συντομία ΑΓΚΜ από τα αρχικά των λέξεων.

Κάθε πίνακας υπέρ του K μπορεί να έρθει σε ΓΚΜ ή σε ΑΓΚΜ μορφή με μία ακολουθία από στοιχειώδεις πράξεις σε γραμμές (γραμμοπράξεις). Με άλλα λόγια ένας πίνακας είναι γραμμοϊσοδύναμος με ένα πίνακα σε ΓΚΜ ή σε ΑΓΚΜ μορφή. Για ένα δοθέντα πίνακα, ο ΑΓΚΜ είναι μοναδικός, αλλά μπορεί να έχει πολλούς ισοδύναμους πίνακες σε ΓΚΜ.

Παράδειγμα 2.4.5 Βρίσκουμε τη ΑΓΚΜ μορφή για τον παρακάτω πίανκα M χρησιμοποιώντας στοιχειώδεις γραμμοπράξεις.

$$M = \begin{bmatrix} 1011 \\ 1010 \\ 1101 \end{bmatrix} \rightarrow \begin{bmatrix} 1011 \\ 0001 \\ 0110 \end{bmatrix} \text{ (προσθέτουμε τη γραμμή 1 στη 2 και στην 3)}$$

$$\rightarrow \begin{bmatrix} 1011 \\ 0110 \\ 0001 \end{bmatrix} \text{ (εναλλάσσουμε τη θέση των γραμμών 2 και 3)}$$

$$\rightarrow \begin{bmatrix} 1010 \\ 0110 \\ 0001 \end{bmatrix} \text{ (προσθέτουμε τη γραμμή 1 στη 2)}$$

Ασκήσεις

2.4.6 Βρείτε τη ΑΓΚΜ μορφή για κάθε έναν από τους τέσσερις πίνακες της άσκησης 2.4.1.

Ο *ανάστροφος* ενός $m \times n$ πίνακα A είναι ο $n \times m$ πίνακας A^T ο οποίος έχει ως $i-$στήλη τη $i-$γραμμή του A. Για παράδειγμα,

$$\text{αν } A = \begin{bmatrix} 1011 \\ 0000 \\ 0110 \end{bmatrix}, \text{ τότε } A^T = \begin{bmatrix} 100 \\ 001 \\ 101 \\ 100 \end{bmatrix}.$$

Θα χρειαστούμε δύο αποτελέσματα σε ανάστροφους πίνακες, των A, B: $(A^T)^T = A$ και $(AB)^T = B^T A^T$.

2.5 Βάσεις για τους $C =< S >$ και $C^{\perp}$

Θα δημιουργήσουμε αλγόριθμους για την εύρεση βάσεων ενός γραμμικού κώδικα και του δυϊκού του. Αυτές οι μέθοδοι θα φανούν πολύ χρήσιμες στη μελέτη των γραμμικών κωδίκων.

Έστω S ένα μη κενό υποσύνολο του K^n. Οι δύο πρώτοι αλγόριθμοι μας δίνουν μία βάση για τον $C =< S >$, το γραμμικό κώδικα που γεννιέται από το S.

Αλγόριθμος 2.5.1 Σχηματίστε τον πίνακα A με γραμές τις λέξεις του S. Χρησιμοποιήστε πράξεις γραμμών για να βρείτε μια ΓΚΜ του A. Τότε οι μη μηδενικές γραμμές της ΓΚΜ σχηματίζουν μία βάση του $C =< S >$.

Ο αλγόριθμος δουλεύει επειδή οι γραμμές του A γεννούν το C και οι στοιχειώδεις πράξεις γραμμών απλά αλλάζουν τη θέση δύο λέξεων ή αντικαθιστούν μία λέξη (γραμμή) με μια άλλη του C δίνοντας ένα νέο σύνολο κωδικολέξεων το οποίο ξαναγεννάει το C. Προφανώς οι μη μηδενικές γραμμές του πίνακα σε ΓΚΜ μορφή είναι γραμμικά ανεξάρτητες.

Παράδειγμα 2.5.2 Βρίσκουμε μία βάση για το γραμμικό κώδικα $C =< S >$ όπου $S = \{11101, 10110, 01011, 11010\}$

$$A = \begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 01011 \\ 00111 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix}.$$

Ο τελευταίος πίνακας είναι μία ΓΚΜ του A. Από τον αλγόριθμο 2.5.1, το $\{11101, 01011, 00111\}$ είναι μία βάση για το $C =< S >$. Άλλη ΓΚΜ του A είναι

$$\begin{bmatrix} 11101 \\ 01100 \\ 00111 \\ 00000 \end{bmatrix},$$

άρα το $\{11101, 01100, 00111\}$ είναι επίσης μία βάση για το $C =< S >$. Ας παρατηρήσουμε ότι ο αγόριθμος 2.5.1 δεν παράγει μία μοναδική βάση για το $< S >$, ούτε οι λέξεις της βάσεως είναι κατ' ανάγκη στοιχεία του δοσμένου συνόλου S.

Ασκήσεις

2.5.3 Χρησιμοποιήστε τον αλγόριθμο 2.5.1 για να βρείτε μία βάση για το $C =< S >$ για κάθε ένα από τα παρακάτω σύνολα S.

(α'). $S = \{010, 011, 111\}$

(β'). $S = \{1010, 0101, 1111\}$

(γ'). $S = \{0101, 1010, 1100\}$

(δ'). $S = \{1000, 0100, 0010, 0001\}$

(ε'). $S = \{11000, 01111, 11110, 01010\}$

(ϛ'). $S = \{10101, 01010, 11111, 00011, 10110\}$

(ζ). $S = \{0110, 1010, 1100, 0011, 1111\}$

(η'). $S = \{111000, 000111, 101010, 010101\}$

(θ'). $S = \{00000000, 10101010, 01010101, 11111111\}$

Αλγόριθμος 2.5.4 Σχηματίστε τον πίνακα A με στήλες τις λέξεις του S. Χρησιμοποιήστε στοιχειώδεις πράξεις γραμμών ώστε να φέρετε τον A σε ΓΚΜ και εντοπίστε τις στήλες οδηγούς στην ΓΚΜ. Τότε οι αρχικές στήλες του A που αντιστοιχούν σε αυτές τις στήλες οδηγούς αποτελούν μία βάση του $C =< S >$.

Αποδεικνύεται στη στοιχειώδη γραμμική άλγεβρα ότι ένα σύνολο από στήλες γραμμικά ανεξάρτητο παραμένει γραμμικά ανεξάρτητο μετά την εφαρμογή στοιχειωδών πράξεων στις γραμμές του πίνακα. Είναι εύκολο να δούμε ότι οι στήλες οδηγοί ενός πίνακα σε ΓΚΜ σχηματίζουν ένα γραμικά ανεξάρτητο σύνολο.

Παράδειγμα 2.5.5 Χρησιμοποιούμε τον αλγόριθμο 2.5.4 για να βρούμε μία βάση του $C =< S >$ για το σύνολο S του παραδείγματος 2.5.2.

$$A = \begin{bmatrix} 1101 \\ 1011 \\ 1100 \\ 0111 \\ 1010 \end{bmatrix} \rightarrow \begin{bmatrix} 1101 \\ 0110 \\ 0001 \\ 0111 \\ 0111 \end{bmatrix} \rightarrow \begin{bmatrix} 1101 \\ 0110 \\ 0001 \\ 0000 \\ 0000 \end{bmatrix}, \text{ το οποίο είναι ΓΚΜ.}$$

Επειδή οι στήλες $1, 2$ και 4 της ΓΚΜ είναι στήλες οδηγοί, ο αλγόριθμος 2.5.4 μας λέει ότι οι στήλες $1, 2$ και 4 του A σχηματίζουν μία βάση του $C =< S >$. Η βάση είναι η $\{11101, 10110, 11010\}$. Ας σημειώσουμε ότι ο αλγόριθμος 2.5.4 έχει την ιδιότητα να παράγει μια βάση για τον $C =< S >$, που όλα τα στοιχεία του, είναι λέξεις του δοθέντος συνόλου S.

Ασκήσεις

2.5.6 Χρησιμοποιείστε τον αλγόριθμο 2.5.4 για να βρείτε μια βάση για το $C =< S >$ για κάθε σύνολο S της άσκησης 2.5.3 και συγκρίνετε τις απαντήσεις.

Τώρα θα δώσουμε έναν αλγόριθμο για να βρούμε μια βάση του δυϊκού κώδικα $C^{\perp}$. Πρόκειται για έναν πολύ χρήσιμο αλγόριθμο για όλη την επόμενη δουλειά μας. Επίσης μπορούμε να παρατηρήσουμε ότι αυτός ο αλγόριθμος μας δίνει μια βάση και για τον C (επειδή εμπεριέχει τον αλγόριθμο 2.5.1).

Αλγόριθμος 2.5.7 Σχηματίστε τον πίνακα A με γραμμές τις λέξεις του S. Χρησιμοπιείστε στοιχειώδεις πράξεις γραμμών για να θέσετε τον A σε ΑΓΚΜ. Έστω G ο $k \times n$ πίνακας που περιέχει όλες τις μη μηδενικές γραμές του ΑΓΚΜ. Έστω X ο $k \times (n-k)$ πίνακας που απομένει από τον G όταν διαγράψουμε τις στήλες οδηγούς του G. Σχηματίστε έναν $n \times (n-k)$ πίνακα H όπως ακολουθεί:

(i) στις γραμμές του H που αντιστοιχούν στις στήλες οδηγούς του G, θέστε με τη σειρά τους, τις γραμμές του X·

(ii) στις υπόλοιπες $n-k$ γραμμές του H, θέστε με τη σειρά τους, τις γραμμές του $(n-k) \times (n-k)$ ταυτοτικού πίνακα I.

Τότε οι στήλες του H αποτελούν μία βάση του $C^{\perp}$.

Ο αλγόριθμος δουλεύει διότι οι $n-k$ στήλες του H είναι γραμμικά ανεξάρτητες, $\dim C^{\perp} = n - \dim C = n-k$ και κάνοντας μία (ταυτόχρονη) κατάλληλη μετάθεση των γραμμών του G και των στηλών του H, $GH = X + X = 0$.

Η παρακάτω περιγραφή του αλγόριθμου 2.5.7 ίσως είναι καλύτερη πρακτικά. Ο πίνακας G περιέχει k στήλες οδηγούς. Μεταθέστε τις στήλες του G έστι ώστε οι στήλες αυτές να έρθουν πρώτες. Οι υπόλοιπες στήλες σχηματίζουν τον πίνακα X. Ονομάστε αυτόν τον πίνακα G'. Οπότε ο αλγόριθμος 2.5.7 περιγράφεται ως:

$$A \rightarrow \begin{bmatrix} G \\ 0 \end{bmatrix} \text{(ΑΓΚΜ)}$$

Μεταθέστε τις στήλες του G και σχηματίστε τον $G' = [I_k, X]$. Σχηματίστε τον πίνακα H' όπως ακολουθεί:

$$H' = \begin{bmatrix} X \\ I_{n-k} \end{bmatrix}.$$

Εφαρμόστε την αντίστροφη μετάθεση που εφαρμόστηκε στις στήλες του G, στις γραμμές του H' για να σχηματιστεί ο H.

Παράδειγμα 2.5.8 Θα χρησιμοποιήσουμε τον αλγόριθμο 2.5.7 για να βρούμε μια βάση για το $C^{\perp}$ για το σύνολο S του παραδείγματος 2.5.2.

$$A = \begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix} \rightarrow \begin{bmatrix} 11010 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix} \rightarrow \begin{bmatrix} 10001 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix},$$

το οποίο είναι ΑΓΚΜ. Τώρα $G = \begin{bmatrix} 100|01 \\ 010|11 \\ 001|11 \end{bmatrix}$, $k = 3$ και $X = \begin{bmatrix} 01 \\ 11 \\ 11 \end{bmatrix}$. Οι στήλες οδηγού του G είναι οι 1, 2 και 3, οπότε οι γραμμές του X τοποθετούνται στις γραμμές 1, 2 και 3 αντίστοιχα, του $5 \times (5-3)$ πίνακα H. Οι υπόλοιες γραμμές του H γεμίζουν με τον 2×2 ταυτοτικό πίνακα. Οπότε

$$H = \begin{bmatrix} 01 \\ 11 \\ 11 \\ --- \\ 10 \\ 01 \end{bmatrix}.$$

Από τον αλγόριθμο 2.5.7 οι στήλες του H σχηματίζουν μία βάση του $C^{\perp}$. Σημειώστε ότι από τον αλγόριθμο 2.5.1, οι γραμμές του G σχηματίζουν μία βάση για το $C = < S >$.

Παράδειγμα 2.5.9 Υποθέτουμε ότι $n = 10$ και ότι έχουμε ένα σύνολο S από λέξεις του K^{10}. Υποθέτουμε ότι η ΑΓΚΜ του πίνακα A για τον αλγόριθμο 2.5.7

έχει μη μηδενικές γραμμές

$$G = \begin{bmatrix} 1\,0\,1\,0\,0\,1\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \end{bmatrix}.$$

Οι στήλες οδηγοί του G είναι οι στήλες $1, 4, 5, 7$ και 9. Μεταθέτουμε τις στήλες του G με την εξής σειρά $1, 4, 5, 7, 9, 2, 3, 6, 8, 10$ (ώστε οι στήλες οδηγοί να έρθουν πρώτες) και σχηματίζουμε τον πίνακα

$$G' = \left[\begin{array}{c|c} 1\,0\,0\,0\,0 & 0\,1\,1\,1\,1 \\ 0\,1\,0\,0\,0 & 0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0 & 0\,0\,0\,1\,0 \\ 0\,0\,0\,1\,0 & 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1 & 0\,0\,0\,0\,1 \end{array}\right].$$

Στη συνέχεια σχηματίζουμε τον πίνακα H' και τελικά επανατοποθετούμε τις γραμμές του H στην κανονική τους σειρά για να σχηματιστεί ο πίνακας H'.

$$H' = \begin{bmatrix} X \\ I \end{bmatrix} = \begin{bmatrix} 0\,1\,1\,1\,1 \\ 0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \end{bmatrix} \begin{matrix} 1 \\ 4 \\ 5 \\ 7 \\ 9 \\ 2 \\ 3 \\ 6 \\ 8 \\ 10 \end{matrix}, H = \begin{bmatrix} 0\,1\,1\,1\,1 \\ 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix}$$

Λόγω του αλγορίθμου 2.5.7, οι στήλες του H σχηματίζουν μία βάση του $C^{\perp}$.

Ασκήσεις

2.5.10 Χρησιμοποιείστε τον αλγόριθμο 2.5.7 για να βρείτε μία βάση για το $C^{\perp}$ για κάθε έναν από τους κώδικες $C =< S >$ όπου

(α'). $S = \{010, 011, 111\}$

(β'). $S = \{1010, 0101, 1111\}$

(γ'). $S = \{0101, 1010, 1100\}$

(δ'). $S = \{1000, 0100, 0010, 0001\}$

(ε'). $S = \{11000, 01111, 11110, 01010\}$

(ϛ'). $S = \{10101, 01010, 11111, 00011, 10110\}$

(ζ'). $S = \{0110, 1010, 1100, 0011, 1111\}$

(η'). $S = \{111000, 000111, 101010, 010101\}$

(θ'). $S = \{00000000, 10101010, 01010101, 11111111\}$

2.5.11 Με το συμβολισμό του αλγόριθμου 2.5.7, εξηγείστε γιατί περιμένουμε να ισχύει $GH = 0$.

2.5.12 Για κάθε ένα από τα παρακάτω σύνολα S, χρησιμοποιείστε τον αλγόριθμο 2.5.7 για να παραχθεί μία βάση B για τον κώδικα $C =< S >$ και μία βάση $B^{\perp}$ για το δυϊκό κώδικα $C^{\perp}$.

(α'). $S = \{000000, 111000, 000111, 111111\}$

(β'). $S = \{1101000, 0110100, 0011010, 0001101, 1000110, 0100011, 1010001\}$

(γ'). $S = \{1111000, 0111100, 0011110, 0001111, 1000111, 1100011, 1110001\}$

(δ'). $S = \{101101110, 011011101, 110110010, 011011110, 111111101\}$

(ε'). $S = \{100100100, 010010010, 111111111, 000000000\}$

(ϛ'). $S = \{001101, 001000, 001111, 000101, 000001\}$

2.6 Γεννήτορες Πίνακες και Κωδικοποίηση

Εφαρμόζουμε το υλικό των τελευταίων παραγράφων για να βρούμε ένα σπουδαίο πίνακα για ένα γραμικό κώδικα και να δούμε πως αυτός ο πίνακας χρησιμοποιείται για να στέλνουμε μηνύματα.

Κατ'αρχάς μερικές αρχικές σημειώσεις. Η *τάξη* ενός πίνακα υπέρ του K είναι το πλήθος των μη μηδενικών γραμμών μιας οποιασδήποτε ΓΚΜ του πίνακα. *Η διάσταση K του κώδικα C* είναι η διάσταση του C, ως υπόχωρος του K^n. Αν ο C έχει μήκος n και απόσταση d, τότε λέμε ότι ο C είναι ένας (n, k, d) γραμμικός κώδικας. Οι τρεις *παράμετροι,* μήκος, διάσταση, απόσταση, αποτελούν σημαντική πληροφορία για τον C.

Αν ο C είναι ένας γραμμικός κώδικας μήκους n και διάστασης k, τότε κάθε πίνακας που οι γραμμές του σχηματίζουν μία βάση του C ονομάζεται *γεννήτορας πίνακας* του C. Σημειώστε ότι ένας γεννήτορας πίνακας για τον C πρέπει να έχει k γραμμές και n στήλες και πρέπει να έχει τάξη k.

Θεώρημα 2.6.1 *Ένας πίνακας G είναι ένας γεννήτορας πίνακας για κάποιο γραμμικό κώδικα C, αν και μόνο αν οι γραμμές του G είναι γραμμικά ανεξάρτητες· δηλαδή αν και μόνο αν η τάξη του G είναι ίση με το πλήθος των γραμμών του G.*

Επειδή οι ισοδύναμοι πίνακες ως προς τις γραμμές έχουν την ίδια τάξη, παίρνουμε το παρακάτω θεώρημα.

Θεώρημα 2.6.2 *Αν G είναι ένας γεννήτορας πίνακας για τον γραμμικό κώδικα C, τότε ένας οποιοσδήποτε ισοδύναμος πίνακας με τον G ως προς τις γραμμές, είναι επίσης ένας γεννήτορας πίνακας για τον C. Ειδικά, κάθε γραμμικός κώδικας έχει έναν γεννήτορα πίνακα σε ΑΓΚΜ.*

Για να βρούμε έναν γεννήτορα πίνακα για ένα γραμμικό κώδικα C, σχηματίζουμε τον πίνακα που οι γραμμές του είναι γραμμές του C. Επειδή $C =< C >$, είτε ο αλγόριθμος 2.5.1 είτε ο αλγόριθμος 2.5.7 μπορεί να χρησιμοποιειθεί για να παραχθεί μία βάση για τον C. Ο πίνακας που οι γραμμές του σχηματίζουν τη βάση είναι ο γεννήτορας πίνακας για τον C.

Παράδειγμα 2.6.3 Βρίσκουμε έναν γεννήτορα πίνακα για τον κώδικα $C = \{0000, 1110, 0111, 1001\}$. Χρησιμοποιώντας τον αλγόριθμο 2.5.1,

$$A = \begin{bmatrix} 0000 \\ 1110 \\ 0111 \\ 1001 \end{bmatrix} \rightarrow \begin{bmatrix} 1110 \\ 0111 \\ 1001 \\ 0000 \end{bmatrix} \rightarrow \begin{bmatrix} 1110 \\ 0111 \\ 0111 \\ 0000 \end{bmatrix} \rightarrow \begin{bmatrix} 1110 \\ 0111 \\ 0000 \\ 0000 \end{bmatrix},$$

οπότε $G = \begin{bmatrix} 1110 \\ 0111 \end{bmatrix}$ είναι ο γεννήτορας πίνακας για τον C. Από τον αλγόριθμο 2.5.7, επειδή η ΑΓΚΜ του A είναι $\begin{bmatrix} 1001 \\ 0111 \\ 0000 \\ 0000 \end{bmatrix}$, ο $G_1 = \begin{bmatrix} 1001 \\ 0111 \end{bmatrix}$ είναι επίσης γεννήτορας πίνακας για τον C.

Ασκήσεις

2.6.4 Προσδιορίστε αν οι παρακάτω πίνακες είναι γεννήτορες πίνακες για κάποιους γραμμικούς κώδικες.

$$A = \begin{bmatrix} 0100111101 \\ 1001011101 \\ 1011001101 \\ 1011011101 \end{bmatrix} \quad B = \begin{bmatrix} 1001101001 \\ 1101000101 \\ 0111001011 \\ 1000010111 \\ 1010001110 \end{bmatrix}$$

2.6.5 Βρείτε τους γεννήτορες πίνακες σε ΑΓΚΜ για κάθε έναν από τους παρακάτω κώδικες.

(α'). $C = \{000, 001, 010, 011\}$

(β'). $C = \{0000, 1001, 0110, 1111\}$

(γ'). $C = \{00000, 11111\}$

(δ'). $C = \{00000, 11100, 00111, 11011\}$

(ε'). $C = \{00000, 11110, 01111, 10001, \}$

(ϛ΄). $C = \{000000, 101010, 010101, 111111\}$

2.6.6 Βρείτε ένα γεννήτορα πίνακα για κάθε έναν από τους παρακάτω κώδικες. Δώστε τη διάσταση του κώδικα.

(α΄). $C = \{000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000\}$

(β΄). $C = \{00000000, 01101111, 11011000, 11111101, 10010010, 00100101, 01001010, 10110111\}$

(γ΄). $C = \{0000000000, 1111100000, 0000011111, 1111111111\}$

2.6.7 Βρείτε έναν γεννήτορα πίνακα για το γραμμικό κώδικα που γεννιέται από καθένα από τα παρακάτω σύνολα. Δώστε τις παραμέτρους (n, k, d) για κάθε κώδικα.

(α΄). $C = \{11111111, 11110000, 11001100, 10101010\}$

(β΄). $C = \{11111100, 11110011, 11001111, 00111111\}$

(γ΄). $C = \{100100100, 010010010, 001001001, 111111111\}$

(δ΄). $C = \{10101, 0101, 11111, 00011, 10110\}$

(ε΄). $C = \{1010, 0101, 1111\}$

(ϛ΄). $C = \{101101, 011010, 110111, 000111, 110000\}$

(ζ΄). $C = \{1001011, 0101010, 1001100, 0011001, 0000111\}$

Έστω C ένας γραμμικός κώδικας μήκους n και διάστασης k. Αν G είναι ένας γεννήτορας πίνακας για τον C και αν n είναι μια λέξη μήκους k γραμμένη ως γραμμή, τότε η $v = uG$ είναι μία λεξη του C, διότι η v είναι γραμμικός συνδυασμός των γραμμών του G, τα οποία σχηματίζουν μια βάση για τον C. Πράγματι, αν $u = (\alpha_1, \alpha_2, ..., \alpha_k)$ και αν

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix},$$

όπου $g_1, g_2, ..., g_k$ είναι γραμμές του G, τότε $v = uG = \alpha_1 g_1, \alpha_2 g_2, ..., \alpha_k g_k$. Από την άλλη μεριά, επειδή κάθε λέξη v του C είναι γραμμικός συνδυασμός των λέξεων της βάσης (των γραμμών του G), τότε έχουμε $v = uG$ για κάποια $u \in K^k$. Ακόμα, αν $u_1 G = u_2 G$, τότε έχουμε $u_1 = u_2$ διότι κάθε λέξη του C γράφεται ως *μοναδικός* γραμμικός συνδυασμός των λέξεων της βάσης. Ώστε καμία λέξη $v = uG$ δεν παράγεται από περισσότερες από μία $u \in K^k$.

Θεώρημα 2.6.8 *Εάν G είναι ένας γεννήτορας πίνακας για ένα γραμμικό κώδικα C μήκους n και διάστασης k, τότε $v = uG$ διατρέχει όλες τις 2^k λέξεις του C καθώς το u διατρέχει όλες τις 2^k λέξεις μήκους k. Έτσι ο C είναι το σύνολο όλων των λέξεων uG, όπου u στο K^k. Ακόμα, $u_1 G = u_2 G$ αν και μόνο αν $u_1 = u_2$.*

Ας παρατηρήσουμε ότι το θεώρημα 2.6.8 μας λέει ότι τα μηνύματα που μπορούν να κωδικοποιηθούν με ένα (n, k, d) γραμμικό κώδικα είναι ακριβώς όλα τα μηνύματα του K^k. Το μήνυμα u κωδικοποιείται ως $v = uG$, οπότε μόνο k ψηφία μιας οποιασδήποτε κωδικολέξης χρησιμοποιούνται για να μεταφέρουν το μήνυμα. Ας σημειώσουμε ότι ο βαθμός πληροφορίας ενός (n, k, d) κώδικα είναι $\log_2(2^k)/n = k/n$.

Παράδειγμα 2.6.9 Έστω C είναι ένας $(5, 3, d)$ γραμμικός κώδικας με γεννήτορα πίνακα τον παρακάτω. Ο βαθμός πληροφορίας του C είναι $k/n = 3/5$. Κάθε μήνυμα u στο K^3 μπορεί να κωδικοποιηθεί. Για παράδειγμα το μήνυμα $u = 101$ κωδικοποιείται ως

$$v = uG = [101] \begin{bmatrix} 1\,0\,1\,1\,0 \\ 0\,1\,0\,1\,1 \\ 0\,0\,1\,0\,1 \end{bmatrix} = 10011.$$

Ασκήσεις

2.6.10 Για κάθε έναν από τους παρακάτω γεννήτορες πίνακες κωδικοποιείστε τα δοθέντα μηνύματα.

(α'). $G = \begin{bmatrix} 1\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0 \\ 0\,0\,1\,0\,1 \end{bmatrix}$

(i) $u = 100$

(ii) $u = 010$

(iii) $u = 111$

(β'). $G = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \end{bmatrix}$

(i) $u = 000$

(ii) $u = 100$

(iii) $u = 111$

(γ'). $G = \begin{bmatrix} 1\,1\,0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,1\,1\,1\,1\,1 \end{bmatrix}$

(i) $u = 1000$

(ii) $u = 1010$

(iii) $u = 0011$

(iv) $u = 1011$

2.6.11 Αντιστοιχείστε γράμματα με λέξεις του K^3 όπως ακολουθεί:

000	100	010	001	110	101	011	111
A	B	E	H	M	R	T	W

Χρησιμοποιώντας τον γεννήτορα πίνακα του παραδείγματος 2.6.9, κωδικοποείστε το μήνυμα BE THERE (αγνοήστε το κενό).

2.6.12 Έστω C ένας κώδικας με γεννήτορα πίνακα

$$G = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,0\,0\,1\,1\,0 \\ 0\,0\,1\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0\,1\,1 \end{bmatrix}.$$

Αντιστοιχίζουμε γράμματα με λέξεις του K^4 όπως ακολουθεί:

0000	1000	0100	0010	0001	1100	1010	1001
A	B	C	D	E	F	G	H
0110	0101	0011	1110	1101	1011	0111	1111
I	J	K	L	M	N	O	P

(α'). Κωδικοποιείστε το μήνυμα HELP.

(β'). Μεταδώστε το μήνυμα HELP υποθέτοντας ότι κατά τη μετάδοση η πρώτη λέξη αποκτά ένα λάθος στην πρώτη θέση, η δεύτερη λέξη δεν έχει λάθη, η τρίτη λέξη ένα λάθος στην έβδομη θέση και η τέταρτη δύο λάθη στην πέμπτη και έκτη θέση.

(γ'). Κωδικοποιείστε το μήνυμα CALL HOME BAMA (Αγνοήστε τα κενά).

2.6.13 Βρείτε το πλήθος των μηνυμάτων που μπορούν να αποσταλούν και τον βαθμό πληροφορίας r, για κάθε έναν από τους γραμμικούς κώδικες στις ασκήσεις 2.6.6 και 2.6.7.

2.7 Πίνακες Ελέγχου Ισοτιμίας (Parity-Check Πίνακες)

Θα αναπτύσουμε έναν άλλο πίνακα που συσχετίζεται με ένα γραμμικό κώδικα και που συνδέεται στενά με έναν γεννήτορα πίνακα. Αυτός ο νέος πίνακας έχει μεγάλη αξία στο σχεδιασμό σχημάτων αποκωδικοποίησης.

Ένας πίνακας H καλείται ένας *parity-check* πίνακας(πίνακας ελέγχου ισοτιμίας) για ένα γραμμικό κώδικα C, αν οι στήλες του H αποτελούν μία βάση για τον δυϊκό κώδικα $C^{\perp}$. Αν ο C έχει μήκος n και διάσταση k, τότε αφού το άθροισμα των διαστάσεων των C και $C^{\perp}$ είναι n, κάθε parity-check πίνακας για τον C πρέπει να έχει n γραμμές και $n-k$ στήλες και τάξη $n-k$. Συγκρίνετε το παρακάτω θεώρημα με το θεώρημα 2.6.1.

Θεώρημα 2.7.1 *Ένας πίνακας H είναι ένας parity-check πίνακας για ένα γραμμικό κώδικα C αν και μόνο αν οι στήλες του H είναι γραμμικά ανεξάρτητες.*

Το παρακάτω θεώρημα περιγράφει ένα γραμμικό κώδικα με όρους του parity-check πίνακά του.

Θεώρημα 2.7.2 *Αν ο H είναι ένας parity-check πίνακας για ένα γραμμικό κώδικα C μήκους n, τότε ο C αποτελείται ακριβώς απ' όλες τις λέξεις v του K^n τέτοιες ώστε $vH = 0$.*

Αν μας δοθεί ο γεννήτορας πίνακας για ένα γραμμικό κώδικα C, τότε μπορούμε να βρούμε έναν parity-check πίνακα του C χρησιμοποιώντας τον αλγόριθμο 2.5.7. Ο parity-check πίνακας είναι ο πίνακας H που κατασκευάστηκε στον αλγόριθμο 2.5.7, επειδή οι στήλες του H σχηματίζουν μια βάση του $C^{\perp}$.

Παράδειγμα 2.7.3 Βρίσκουμε έναν parity-check πίνακα για τον κώδικα $C = \{0000, 1110, 0111, 100\}$ του παραδείγματος 2.6.3. Εκεί είχαμε βρει ότι ο

$$G_1 = \begin{bmatrix} 10\,01 \\ 01\,11 \end{bmatrix} = [IX]$$

είναι ο γεννήτορας πίνακας του C, ο οποίος είναι σε ΑΓΚΜ. Από τον αλγόριθμο 2.5.7, φτιάχνουμε τον H:

$$H = \begin{bmatrix} X \\ I \end{bmatrix} = \begin{bmatrix} 01 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

ο οποίος είναι ένας parity-check πίνακας για τον C. Ας παρατηρήσουμε ότι $vH = 00$ για όλες τις λέξεις v του C.

Ασκήσεις

2.7.4 Βρείτε έναν parity-check πίνακα για καθέναν από τους παρακάτω κώδικες.

(α'). $C = \{000, 001, 010, 011\}$

(β'). $C = \{0000, 1001, 0110, 1111\}$

(γ'). $C = \{00000, 11111\}$

(δ'). $C = \{00000, 11100, 00111, 11011\}$

(ε'). $C = \{00000, 11110, 01111, 10001, \}$

(ϛ'). $C = \{000000, 101010, 010101, 111111\}$

2.7.5 Βρείτε έναν parity-check πίνακα για καθέναν από τους παρακάτω κώδικες (οι γεννήτορες πίνακες έχουν κατασκευαστεί στις ασκήσεις 2.6.6 και 2.6.7).

(α΄). $C = \{000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000\}$

(β΄). $C = \{00000000, 01101111, 11011000, 11111101, 10010010, 00100101, 01001010, 10110111\}$

(γ΄). $C = \{0000000000, 1111100000, 0000011111, 1111111111\}$

(δ΄). $C =< S >, S = \{11111111, 11110000, 11001100, 10101010\}$

(ε΄). $C =< S >, S = \{11111100, 11110011, 11001111, 00111111\}$

(ϛ΄). $C =< S >, S = \{100100100, 010010010, 001001001, 111111111\}$

(ζ΄). $C =< S >, S = \{10101, 0101, 11111, 00011, 10110\}$

(η΄). $C =< S >, S = \{1010, 0101, 1111\}$

(θ΄). $C =< S >, S = \{101101, 011010, 110111, 000111, 110000\}$

(ι΄). $C =< S >, S = \{1001011, 0101010, 1001100, 0011001, 0000111\}$

Τώρα θα εκφράσουμε τη σχέση μεταξύ ενός γεννήτορα πίνακα και του parity-check πίνακα ενός γραμμικού κώδικα και τη σχέση μεταξύ αυτών των πινάκων με ένα γραμμικό κώδικα και του δυϊκού του κώδικα.

Θεώρημα 2.7.6 *Οι πίνακες G και H είναι γεννήτορες και parity-check πίνακες αντίστοιχα, για κάποιο γραμμικό κώδικα C αν και μόνο αν*

(i) οι γραμμές του G είναι γραμμικά ανεξάρτητες,

(ii) οι στήλες του H είναι γραμμικά ανεξάρτητες,

(iii) το πλήθος των γραμμών του G μαζί με το πλήθος των στηλών του H είναι ίσο με το πλήθος των στηλών του G το οποίο είναι ίσο με το πλήθος των γραμμών του H και

(iv) $GH = 0$.

Θεώρημα 2.7.7 *H είναι ένας parity-check πίνακας του C αν και μόνο αν ο H^T είναι ένας γεννήτορας πίνακας για τον $C^\perp$.*

Το θεώρημα 2.7.7 βγαίνει από το θεώρημα 2.7.6 και το γεγονός ότι

$$H^T G^T = (GH)^T = 0.$$

Δοθέντος ενός γεννήτορα πίνακα ή parity-check πίνακα του C ή του $C^\perp$, μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο 2.5.7 και το θεώρημα 2.7.7 για να βρούμε τους άλλους τρεις πίνακες. Το παρακάτω διάγραμμα δείχνει πως μπορεί να γίνει αυτό.

$$\begin{array}{ccc} H_{C_\perp} & \xleftarrow{\text{Αλγόριθμος 2.5.7}} & G_{C_\perp} \\ \text{Αναστροφή}\downarrow & & \uparrow\text{Αναστροφή} \\ G_C & \xrightarrow{\text{Αλγόριθμος 2.5.7}} & H_C \end{array}$$

Παράδειγμα 2.7.8 Έστω C ένας γραμμικός κώδικας με parity-check πίνακα

$$H = \begin{bmatrix} 11 \\ 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = \begin{bmatrix} X \\ I \end{bmatrix}.$$

(α΄). Τότε ο γεννήτορας πίνακας του $C_\perp$ είναι ο

$$H^T = \begin{bmatrix} 11010 \\ 11101 \end{bmatrix}.$$

(β΄). Η ΑΓΚΜ του H^T είναι $H^T = \begin{bmatrix} 11010 \\ 00111 \end{bmatrix}$, οπότε από τον αλγόριθμο 2.5.7, ένας parity-check πίνακας για τον $C^\perp$ είναι

$$\begin{bmatrix} 110 \\ 100 \\ 011 \\ 010 \\ 001 \end{bmatrix}.$$

(γ΄). Από τη μορφή του H, εύκολα έχουμε ότι ο

$$G = \begin{bmatrix} 100 & 11 \\ 010 & 11 \\ 001 & 11 \end{bmatrix} = [I, X]$$

είναι ένας γεννήτορας πίνακας του C. Αυτό το διαπιστώνουμε εάν χρησιμοποιήσουμε τον αλγόριθμο 2.5.7 προς τα πίσω. Οπότε, από το θεώρημα 2.7.7 ο $G^\perp$ είναι επίσης parity-check πίνακας για τον $C^\perp$.

$$G^T = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 1\,1\,0 \\ 1\,1\,1 \end{bmatrix}.$$

Ασκήσεις

2.7.9 Σε κάθε μία περίπτωση δίνεται ένας parity-check πίνακας για ένα γραμμικό κώδικα C. Βρείτε

(i) έναν γεννήτορα πίνακα για τον $C^\perp$ και

(ii) έναν γεννήτορα πίνακα για τον C.

(α'). $H = \begin{bmatrix} 1\,0\,0 \\ 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}$

(β'). $H = \begin{bmatrix} 0\,1 \\ 1\,0 \\ 0\,1 \\ 1\,0 \\ 0\,1 \end{bmatrix}$

(γ'). $H = \begin{bmatrix} 1\,1\,1 \\ 1\,1\,0 \\ 1\,0\,1 \\ 0\,1\,1 \\ 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}$.

2.7.10 Αφού γράψετε όλες τις λέξεις του δυϊκού κώδικα $C^{\perp}$ του κώδικα $C = \{00000, 11111\}$, βρείτε τον γεννήτορα πίνακα και τον parity-check πίνακα για τον $C^{\perp}$.

2.7.11 Για κάθε κώδικα C από τους παρακάτω βρείτε τη διάσταση του C, τη διάσταση του $C^{\perp}$, τις διαστάσεις (μήκος $\times$ πλάτος) του γεννήτορα πίνακα και του parity-check του C και του $C^{\perp}$, το πλήθος των λέξεων του C και του $C^{\perp}$ και τέλος τους βαθμούς πληροφορίας του C και του $C^{\perp}$.

(α'). Ο C έχει μήκος $n = 2^t - 1$ και διάσταση t.

(β'). Ο C έχει μήκος $n = 23$ και διάσταση 11.

(γ'). Ο C έχει μήκος $n = 15$ και διάσταση 8.

2.8 Ισοδύναμοι Πίνακες

Κάθε $k \times n$ πίνακας G με $k < n$ του οποίου k στήλες σχηματίζουν $k \times k$ ταυτοτικό πίνακα I_k, έτσι ώστε:

$$G = [I_k, X],$$

αυτόματα έχει γραμμικώς ανεξάρτητες γραμμές και είναι σε ΑΓΚΜ. Οπότε ο G είναι ένας γεννήτορας πίνακας, για κάποιο γραμμικό κώδικα μήκους n και διάστασης k. Ένας τέτοιος γεννήτορας πίνακας λέγεται ότι είναι σε *κανονική μορφή (standard form)* και ο κώδικας C που γεννιέται από τον G ονομάζεται *συστηματικός κώδικας (systematic code)*.

Δεν έχουν όλοι οι γραμμικοί κώδικες ένα γεννήτορα πίνακα σε κανονική μορφή. Για παράδειγμα ο κώδικας που ορίζεται από τον γεννήτορα πίνακα στην παρακάτω άσκηση έχει πέντε ακόμα διαφορετικούς γεννήτορες πίνακες. Κανένας απ' αυτούς δεν είναι σε κανονική μορφή, όπως δεν είναι ούτε ο G.

Ασκήσεις

2.8.1 Βρείτε τους άλλους πέντε γεννήτορες πίνακες για τον κώδικα που γεννιέται από τον

$$G = \begin{bmatrix} 1\,0\,0 \\ 0\,0\,1 \end{bmatrix}.$$

Είναι όμως προτιμότερο να χρησιμοποιούμε κώδικες που έχουν γεννήτορες πίνακες σε κανονική μορφή. Ένας λόγος είναι ότι εάν ο γραμμικός κώδικας C έχει γεννήτορα πίνακα G σε κανονική μορφή, $G = [I, X]$ τότε ο αλγόριθμος 2.5.7 παράγει αμέσως τον πίνακα

$$H = \begin{bmatrix} X \\ I \end{bmatrix}$$

ο οποίος είναι ένας parity-check πίνακας του C.

Από το θεώρημα 2.6.8 κάθε κωδικολέξη v σε ένα γραμμικό κώδικα C μήκους n και διάστασης k είναι ίση με uG για κάποια μοναδική λέξη u του K^k, όπου ο G είναι ένας γεννήτορας πίνακας για τον C. Σκεφτόμαστε τη λέξη u μήκους k ως το μήνυμα το οποίο θα αποσταλεί. Αντί όμως της u, στέλνουμε φυσικα την κωδικολέξη $v = uG$. Εάν η ΑΜΠ σωστά συμπεραίνει ότι η $v = uG$ αποστάλθηκε, τότε ο παραλήπτης του μηνύματος πρέπει με κάποιο τρόπο να αποκαλύψει το αρχικό μήνυμα u από το uG. Εάν ο G είναι σε κανονική μορφή, τότε είναι τετριμένο να αποκαλύψουμε το u από το uG. Διότι σ' αυτήν την περίπτωση

$$v = uG = u[I\,X] = [uI\,uX] = [u\,uX].$$

Οπότε δείξαμε το παρακάτω θεώρημα, το οποίο μας τονίζει ένα σημαντικό πλεονέκτημα που έχει ένας γεννήτορας πίνακας σε κανονική μορφή.

Θεώρημα 2.8.2 *Εάν C είναι ένας γραμμικός κώδικας μήκους n και διάστασης k με γεννήτορα πίνακα του G σε κανονική μορφή, τότε τα πρώτα k ψηφία της κωδικολέξης $v = uG$ σχηματίζουν τη λέξη u του K^k.*

Παράδειγμα 2.8.3 Εάν

$$G = \left[\begin{array}{c} 1\,0\,0\,0 \,|\, 1\,0\,1 \\ 0\,1\,0\,0 \,|\, 1\,0\,0 \\ 0\,0\,1\,0 \,|\, 1\,1\,0 \\ 0\,0\,0\,1 \,|\, 0\,1\,1 \end{array}\right] = [I_4\,X]$$

και εάν το μήνυμα είναι $u = 0111$, τότε $uG = 0111001 = [u001]$. Και έαν $u = 1011$, τότε $uG = 1011000$.

Ασκήσεις

2.8.4 Έστω C γεννήτορας πίνακας του παραδείγματος 2.8.3. Κωδικοποιείστε καθένα από τα παρακάτω μηνύματα u και παρατηρήστε ότι τα πρώτα 4 ψηφία στην κωδικολέξη που προκύπτει σχηματίζουν το μήνυμα u.

(α'). $u = 1111$

(β'). $u = 1011$

(γ'). $u = 0000$

2.8.5 Βρείτε μία μέθοδο να ανακαλύπτετε τη u από τον uG, εάν ο G δεν είναι σε κανονική μορφή.

2.8.6 Εάν ένας γραμμικός κώδικας C έχει γεννήτορα πίνακα τον

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

αποκαλύψτε τη u από την $u = uG = 0000101$.

Με τις υποθέσεις του θεωρήματος 2.8.2 τα πρώτα k ψηφία της κωδικολέξης $v = uG$ λέγονται *ψηφία πληροφορίας (information digits),* αφού ουσιαστικά περιέχουν το μήνυμα u, ενώ τα υπόλοιπα $n - k$ ψηφία της $v = uG$ λέγονται *πλεονάζοντα ή parity-check ψηφία (redundancy ή parity-check digits).*

Έχοντας όλα αυτά τα πλεονεκτήματα για ένα γραμμικό κώδικα με γεννήτορα πίνακα σε κανονική μορφή, τι μπορεί να γίνει εάν ένας κώδικας C δεν έχει γεννήτορα πίνακα σε κανονική μορφή; Ας θεωρήσουμε τον κώδικα C με γεννήτορα πίνακα τον G στην άσκηση 2.8.1. Όπως δείξαμε στην άσκηση 2.8.1 ο C δεν έχει γεννήτορα πίνακα σε κανονική μορφή. Ας υποθέσουμε, γι' αυτό το παράδειγμα, ότι αποφασίζουμε να αναδιατάξουμε τα ψηφία των κωδικολέξεων και να μεταδίδουμε τα ψηφία με τη σειρά «πρώτο, τρίτο, δεύτερο» αντί για «πρώτο, δεύτερο, τρίτο». Οι τέσσερις λέξεις του C έχουν λοιπόν μετασχηματιστεί σε τέσσερις λέξεις στο νέο κώδικα C' όπως φαίνεται παρακάτω:

$$C = \{000, 100, 001, 101\}$$

$$C' = \{000, 100, 010, 110\}$$

Σημειώστε ότι ο C', παρ' όλο που είναι διαφορετικός κώδικας από τον C, έχει πολλές κοινές ιδιότητες με τον C. Για παράδειγμα, είναι και οι δυο γραμμικοί, έχουν το ίδιο μήκος ίσο με 3, διάσταση 2 και απόσταση 1. Όμως ο C' έχει ένα πλεονέκτημα από τον C, δηλαδή ο C' έχει έναν γεννήτορα πίνακα σε κανονική μορφή. Παρατηρήστε ότι ο G' προήλθε από τον G με μετατόπιση της δεύτερης

και της τρίτης στήλης, όπως ακριβώς ο C' προήλθε από τον C με μετατόπιση του δεύτερου και τρίτου ψηφίου.

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

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

Εάν ο C είναι ένας οποιοσδήποτε μπλοκ κώδικας μήκους n, τότε μπορούμε πάντοτε να βρούμε ένα νέο μπλοκ κώδικα C μήκους n, διαλέγοντας μια συγκεκριμένη μετατόπιση των n ψηφίων και στη συνέχεια αναδιατάσσοντας κατάλληλα κάθε λέξη του C κατά τον επιλεγμένο τρόπο. Ο κώδικας C', που προκύπτει, λέγεται *ισοδύναμος* του C.

Παράδειγμα 2.8.7 Εάν $n = 5$ και επιλέξουμε να αναδιατάξουμε τα ψηφία με τη σειρά 2, 1, 4, 5, 3, τότε ο κώδικας

$$C = \{11111, 10111, 00111, 00011, 00001\}$$

είναι ισοδύναμος με τον κώδικα

$$C' = \{11111, 10111, 00111, 00110, 00010\}.$$

(Σημειώστε ότι ο C και ο C' δεν είναι γραμμικοί).

Θεώρημα 2.8.8 *Κάθε γραμμικός κώδικας C είναι ισοδύναμος με ένα γραμμικό κώδικα C' που έχει γεννήτορα πίνακα σε κανονική μορφή.*

Απόδειξη: Έστω ότι ο G είναι γεννήτορας πίνακας του C και το θέτουμε σε ΑΓΚΜ. Αναδιατάσσουμε τις στήλες του ΑΓΚΜ έτσι ώστε οι στήλες οδηγοί να έρθουν πρώτες ώστε να σχηματίσουμε τον ταυτοτικό πίνακα. Το αποτέλεσμα είναι ένας πίνακας G' σε κανονική μορφή, το οποίο είναι ο γεννήτορας πίνακας για έναν κώδικα C' ισοδύναμο με τον C.

Παράδειγμα 2.8.9 Ο πίνακας

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

είναι ένας γεννήτορας πίνακας ΑΓΚΜ με τις στήλες 2, 4, 5, 6 και 9 ως οδηγούς. Αναδιατάσσοντας τις στήλες με τη σειρά 2, 4, 5, 6, 9, 1, 3, 7, 8 παράγεται ο πίνακας

$$G' = \begin{bmatrix} 1\,0\,0\,0\,0 & 0\,1\,0\,1 \\ 0\,1\,0\,0\,0 & 0\,0\,1\,1 \\ 0\,0\,1\,0\,0 & 0\,0\,0\,1 \\ 0\,0\,0\,1\,0 & 0\,0\,1\,0 \\ 0\,0\,0\,0\,1 & 0\,0\,0\,0 \end{bmatrix} = [I\,X],$$

ο οποίος είναι ένας γεννήτορας πίνακας σε κανονική μορφή για έναν κώδικα ισοδύναμο με τον κώδικα που γεννιέται από τον G.

Ασκήσεις

2.8.10 Βρείτε ένα συστηματικό κώδικα C' ισοδύναμο με τον δοθέντα κώδικα C. Ελέγξτε ότι οι C και C' έχουν το ίδιο μήκος, διάσταση και απόσταση.

(α'). $C = \{00000, 10110, 10101, 00011\}$

(β'). $C = \{00000, 11100, 00111, 11011\}$.

2.8.11 Βρείτε έναν γεννήτορα πίνακα G σε κανονική μορφή για έναν κώδικα ισοδύναμο με τον κώδικα που έχει τον παρακάτω γεννήτορα G.

(α) $G = \begin{bmatrix} 101010 \\ 011000 \\ 110100 \\ 101011 \end{bmatrix}$ (β) $G = \begin{bmatrix} 111000000 \\ 000111000 \\ 000111111 \end{bmatrix}$

2.8.12 Βρείτε έναν γεννήτορα πίνακα G' σε κανονική μορφή για έναν κώδικα C' ισοδύναμο με τον κώδικα C με τον παρακάτω parity-check πίνακα H.

(α) $H = \begin{bmatrix} 110 \\ 100 \\ 011 \\ 010 \\ 001 \end{bmatrix}$ (β) $H = \begin{bmatrix} 100 \\ 111 \\ 010 \\ 110 \\ 101 \\ 001 \\ 011 \end{bmatrix}$.

2.8.13 Αποδείξτε ότι οι ισοδύναμοι γραμμικοί κώδικες έχουν πάντα το ίδιο μήκος, διάσταση και απόσταση.

2.8.14 Ελέγξτε ποια από τα παρακάτω ζευγάρια πινάκων G_1 και G_2 παράγουν ισοδύναμους κώδικες.

(α)

$$G_1 = \begin{bmatrix} 1100 \\ 0110 \\ 0011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1001 \\ 0101 \\ 0011 \end{bmatrix}$$

(β)

$$G_1 = \begin{bmatrix} 110000 \\ 001100 \\ 000011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 111111 \\ 011011 \\ 001001 \end{bmatrix}$$

(γ)

$$G_1 = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1011000 \\ 0101100 \\ 0010110 \\ 0001011 \end{bmatrix}.$$

2.9 Απόσταση ενός γραμμικού κώδικα

Έχουμε δει ότι η απόσταση ενός γραμμικού κώδικα είναι το ελάχιστο βάρος των μη μηδενικών κωδικολέξεων. Η απόσταση ενός γραμμικού κώδικα μπορεί επίσης να βρεθεί από τον parity-check πίνακα ενός κώδικα.

Θεώρημα 2.9.1 *Έστω H είναι ένας parity-check πίνακας για ένα γραμμικό κώδικα C. Τότε ο C έχει απόσταση ίση με d αν και μόνο αν κάθε σύνολο από $d-1$ γραμμές του H είναι γραμμικά ανεξάρτητο και τουλάχιστον ένα σύνολο απο d γραμμές του H είναι γραμμικά εξαρτημένο.*

Η ιδέα είναι ότι αν η v είναι μια λέξη, τότε το vH είναι ένας γραμμικός συνδυασμός από $wt(v)$ ακριβώς γραμμές του H. Οπότε εάν η v ανήκει στον C και $wt(v) = d$, τότε επειδή $vH = 0$, μερικές από τις γραμμές του H, πλήθους d είναι γραμμικά εξαρτημένες . Αντίστροφα αν $vH = 0$ τότε η v είναι μία κωδικολέξη με $wt(v) \geq d$.

Παράδειγμα 2.9.2 Έστω C είναι ο γραμμικός κώδικας με parity-check πίνακα

$$H = \begin{bmatrix} 110 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Μετά από έλεγχο διαπιστώνουμε ότι οι δύο γραμμές του H δεν έχουν άθροισμα 000, οπότε κάθε δύο γραμμές του H είναι γραμμικά ανεξάρτητες, όμως οι γραμμές 1, 3 και 4 για παράδειγμα, έχουν άθροισμα 000 και έτσι είναι γραμμικά εξαρτημένες. Οπότε $d - 1 = 2$, έτσι η απόσταση του C είναι $d = 3$.

Ασκήσεις

2.9.3 Βρείτε τον κώδικα C στο παράδειγμα 2.9.2. Υπολογίστε το βάρος κάθε κωδικολέξης και βεβαιώστε ότι ο C έχει απόσταση 3.

2.9.4 Βρείτε την απόσταση του γραμμικού κώδικα C για καθέναν από τους παρακάτω parity-check πίνακες. Χρησιμοποιείστε το θεώρημα 2.9.1 και στη συνέχεια δοκιμάστε την απάντησή σας βρίκοντας το $wt(v)$ για κάθε v του C.

(α) $H = \begin{bmatrix} 0111 \\ 1110 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$ (β) $H = \begin{bmatrix} 1110 \\ 1101 \\ 1011 \\ 0111 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$ (γ) $H = \begin{bmatrix} 1101 \\ 1011 \\ 1110 \\ 1110 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$.

2.9.5 Βρείτε από το θεώρημα 2.9.1, την απόσταση του γραμμικού κώδικα με τους παρακάτω γεννήτορες πίνακες.

(α) $G = \begin{bmatrix} 111000000 \\ 000111000 \\ 111111111 \end{bmatrix}$ (β) $G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$.

2.10 Σύμπλοκα

Σ' αυτήν την ενότητα θεωρούμε μία έννοια η οποία θα είναι χρήσιμη στην αποκωδικοποίηση ενός γραμμικού κώδικα, την οποία θα δούμε στην επόμενη ενότητα.

Εάν ο C είναι ένας γραμμικός κώδικας μήκους n, και εάν u είναι μία λέξη μήκους n, ορίζουμε το *σύμπλοκο του C που ορίζεται από τη u* να είναι το σύνολο όλων των λέξεων της μορφής $v + u$ καθώς το v διατρέχει όλες τις λέξεις του C. Συμβολίζουμε αυτό το σύμπλοκο με $C + u$. Έτσι

$$C + u = \{v + u | v \in C\}.$$

Παράδειγμα 2.10.1 Έστω $C = \{000, 111\}$ και έστω $u = 101$. Τότε

$$C + 101 = \{000 + 101, 111 + 101\} = \{101, 010\}.$$

Ας παρατηρήσουμε επίσης ότι

$$C + 111 = \{000 + 111, 111 + 111\} = \{111, 000\} = C$$

και

$$C + 010 = \{000 + 010, 111 + 010\} = \{010, 101\} = C + 101.$$

Ασκήσεις

2.10.2 Γράψτε τα υπόλοιπα από τα σύμπλοκα του $C = \{000, 111\}$. Σημειώστε ότι υπάρχουν οκτώ περιπτώσεις για τα σύμπλοκα του C, μία για κάθε λέξη στον K^3, αλλά μονάχα τέσσερα από αυτά τα σύμπλοκα είναι διαφορετικά.

Εάν ο C είναι ένας γραμμικός κώδικας μήκους n, τότε ίσως κάποιος σκεφτεί ότι υπάρχουν 2^n διαφορετικά σύμπλοκα $C + u$ του C, ένα για κάθε μία από τις 2^n διαφορετικές λέξεις u μήκους n. Όπως δείχνει το παράδειγμα 2.10.1 και η άσκηση 2.10.2, αυτό είναι σχεδόν ατίθασο. Είναι πολύ πιθανό το $C + u_1$ να είνα ταυτόσημο με το $C + u_2$, με $u_1 \neq u_2$.

Το επόμενο θεώρημα περιέχει αρκετές σπουδαίες και χρήσιμες προτάσεις για τα σύμπλοκα. Μία προσεκτική μελέτη των παραδειγμάτων που ακολουθούν το θεώρημα θα βοηθήσει να γίνουν οι προτάσεις αυτές κατανοητές.

Θεώρημα 2.10.3 *Έστω C ένας γραμμικός κώδικας μήκους n. Έστω u και v είναι λέξεις μήκους n.*

1) *Εάν η u ανήκει στο σύμπλοκο $C+u$, τότε $C+u=C+v$· δηλαδή κάθε λέξη σ' ένα σύμπλοκο ορίζει μονοσήμαντα αυτό το σύμπλοκο.*

2) *Η λέξη u ανήκει στο σύμπλοκο $C+u$.*

3) *Εάν $u+v$ ανήκουν στο C, τότε οι u και v ανήκουν στο ίδιο σύμπλοκο.*

4) *Εάν $u+v$ δεν ανήκουν στο C, τότε οι u και v ανήκουν σε διαφορετικό σύμπλοκο.*

5) *Κάθε λέξη του K^n περιέχεται μόνο σε ένα σύμπλοκο του C· δηλαδή είτε $C+u=C+v$, ή $C+u$ και $C+v$ δεν έχουν κοινές λέξεις.*

6) *$|C+u|=|C|$· δηλαδή το πλήθος των λέξεων σε ένα σύμπλοκο του C είναι ίσο με το πλήθος των λέξεων του κώδικα C.*

7) *Εάν ο C έχει διάσταση k, τότε υπάρχουν ακριβώς 2^{n-k} διαφορετικά σύμπλοκα του C και κάθε σύμπλοκο περιέχει ακριβώς 2^k λέξεις.*

8) *Ο κώδικας C είναι ο ίδιος με ένα από τα σύμπλοκά του.*

Παράδειγμα 2.10.4 Θα καταγράψουμε τα σύμπλοκα του κώδικα

$$C=\{0000,1011,0101,1110\}.$$

Κατ'αρχάς, ο ίδιος ο C είναι ένα σύμπλοκο από το (8) του θεωρήματος 2.10.3. (Οι αριθμοί στις παρενθέσεις υποδεικνύουν στον αναγνώστη να ανατρέχει στα αντίστοιχα μέρη του θεωρήματος 2.10.3) Κάθε λέξη του C θα ορίζει το σύμπλοκο C από (1) και (5), οπότε διαλέγουμε μια λέξη u στον k^4 που δεν ανήκει στον C. Για την αποκωδικοποίηση αργότερα, θα βοηθήσει να επιλέξουμε τη u με το μικρότερο δυνατό βάρος. Οπότε ας πάρουμε τη $u=1000$. Οπότε παίρνουμε το σύμπλοκο

$$C+1000=\{1000,0011,1101,0110\}$$

προσθέτοντας τη 1000 σε κάθε λέξη του C. Σημειώστε ότι η $u=1000$ ανήκει στο σύμπλοκο $C+u=C+1000$. Τώα επιλέγουμε άλλη λέξη, ελάχιστου βάρους, στο K^4 που δεν ανήκει στο C ή στο $C+1000$, ας πούμε 0100. Σχηματίζουμε ένα άλλο σύμπλοκο

$$C+0100=\{0100,1111,0001,1010\}.$$

Επαναλαμβάνοντας τη διαδικασία με 0010 παράγεται το σύμπλοκο

$$C+0010=\{0010,1001,0111,1100\}.$$

Ο κώδικας C έχει διάσταση $k=2$. Έχουμε καταγράψει $2^{n-k}=2^{4-2}=2^2=4$ σύμπλοκα, το καθένα με $2^k=2^2=4$ λέξεις και κάθε λέξη του K^4 εμφανίζεται σ' ένα μόνο σύμπλοκο. Επίσης ας παρατηρήσουμε ότι η $0001+1010=1011$ ανήκει στο C, έτσι οι 0001 και 010 ανήκουν στο ίδιο σύμπλοκο, δηλαδή το $C+0100$ (δείτε (3». Από την άλλη μεριά η $0100+0010=0110$ δεν ανήκει στο C και οι 0100 και 0010 ανήκουν σε διαφορετικά σύμπλοκα (δείτε (4)).

Παράδειγμα 2.10.5 Θα καταγράψουμε τα σύμπλοκα του γραμμικού κώδικα C με γεννήτορα πίνακα τον $G = \begin{bmatrix} 100110 \\ 010011 \\ 001111 \end{bmatrix}$.

000000	100000	010000	001000
100110	000110	110110	101110
010011	110011	000011	011011
001111	101111	011111	000111
110101	010101	100101	111101
101001	001001	111001	100001
011100	111100	001100	010100
111010	011010	101010	110010

000100	000010	000001	000101
100010	100100	100111	100011
010111	010001	010011	010110
001011	001101	001110	001010
110001	110111	110100	110000
101101	101011	101000	101100
011000	011110	011101	011001
111110	111000	111000	111111

Καταγράψαμε τα 8 σύμπλοκα. Το πρώτο είναι ο ίδιος ο C. Η λέξη u που χρησιμοποιήθηκε για να σχηματιστεί η $C + u$ βρίσκεται στην κορυφή κάθε σύμπλοκου, αφού $u = 0 + u$ και ελέγχθηκε όπως στο παράδειγμα 2.10.4.

Ασκήσεις

2.10.6 Καταγράψτε τα σύμπλοκα σε καθέναν από τους παρακάτω γραμμικούς κώδικες.

(α'). $C = \{0000, 1001, 0101, 1100\}$

(β'). $C = \{0000, 1010, 1101, 0111\}$

(γ'). $C = \{00000, 10100, 01011, 11111\}$

(δ'). $C = \{0000\}$.

2.10.7 Καταγράψτε τα σύμπλοκα καθενός από τους γραμμικούς κώδικες με γεννήτορα πίνακα τον παρακάτω.

(α'). $G = \begin{bmatrix} 111000 \\ 001110 \\ 100011 \end{bmatrix}$

(β'). $G = \begin{bmatrix} 101010 \\ 010101 \end{bmatrix}$

(γ'). $G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$

(δ'). $G = \begin{bmatrix} 10001 \\ 01001 \\ 00101 \\ 00011 \end{bmatrix}$

(ε'). $G = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$

(ϛ'). $G = [1111]$.

2.10.8 Καταγράψτε τα σύμπλοκα του κώδικα που έχει τον παρακάτω parity-check πίνακα.

(α'). $H = \begin{bmatrix} 10 \\ 11 \\ 10 \\ 01 \end{bmatrix}$

(β'). $H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}$

(γ'). $H = \begin{bmatrix} 100 \\ 010 \\ 010 \\ 001 \\ 001 \\ 001 \end{bmatrix}$.

2.10.9 Αποδείξτε το θεώρημα 2.10.3.

2.11 Η ΑΜΠ για γραμμικούς κώδικες

Ένας από τους σκοπούς μας είναι να σχεδιάσουμε κώδικες οι οποίοι θα μας επιτρέπουν εύκολη και γρήγορη αποκωδικοποίηση της παραληφθείσας λέξης. Γραμμικοί κώδικες πράγματι αποδέχονται μια πιο αποτελέσματική μέθοδο για να διαχειριστούμε την ΑΜΠ χωρίς τη χρήση ενός ΗΑΜΠ πίνακα. Θα περιγράψουμε

μια διαδικασία όμοια για την ΠΑΜΠ και την ΗΑΜΠ για ένα γραμμικό κώδικα. Ο parity-check πίνακας και τα σύμπλοκα ενός κώδικα παίζουν βασικούς ρόλους στη διαδικασία αποκωδικοποίησης.

Έστω C ένας γραμμικός κώδικας. Υποθέτουμε ότι η κωδικολέξη v του C μεταδίδεται και ότι η λέξη w παραλαμβάνεται, με αποτέλεσμα το υπόδειγμα λάθους $u = v + u$. Τότε $w + u = v$ ανήκει στο C, έτσι *το υπόδειγμα λάθους της u και η παραήθφείσα λέξη w είναι μέσα στο ίδιο σύμπλοκοτου C*, σύμφωνα με το (3) του θεωρήματος 2.10.3.

Επειδή τα υποδείγματα λάθους με μικρό βάρος είναι πιο πιθανό να εμφανιστούν, ας δούμε εδώ πως η ΑΜΠ δουλεύει για ένα γραμμικό κώδικα C. Έχοντας λάβει τη λέξη w, επιλέγουμε μια λέξη u, ελαχίστου βάρους, στο σύμπλοκο $C + w$ και συμπεραίνουμε ότι η $v = w + u$ έχει αποσταλεί.

Παράδειγμα 2.11.1 Έστω $C = \{0000, 1011, 0101, 1110\}$. Τα σύμπλοκα του C (δες παράδειγμα 2.10.4) είναι

0000	1000	0100	0010
1011	0011	1111	1001
0101	1101	0001	0111
1110	0110	1010	1100

Ας υποθέσουμε ότι παραλήφθηκε η $w = 1101$. Το σύμπλοκο $C + w = C + 1101$, που περιέχει την w είναι η δεύτερη στήλη. Η λέξη με το μικρότερο βάρος σ' αυτό το σύμπλοκο είναι η $u = 1000$, την οποία παίρνουμε και ως υπόδειγμα λάθους. Συνεπώς συμπεραίνουμε ότι η $v = w+u = 1101+1000 = 0101$ είναι η κωδικολέξη που πολύ πιθανό μας αποστάλθηκε.

Ασκήσεις

2.11.2 Έστω C είναι ο κώδικας του παραδείγματος 2.10.5. Χρησιμοποιώντας την παραπάνω διαδικασία για την ΠΑΜΠ που μόλις περιγράψαμε αποκωδικοποιείστε καθεμιά από τις παρακάτω λέξεις.

(α'). 000011

(β'). 001001

(γ'). 001101

(δ'). 010110

(ε'). 110101

(ϛ'). 001010.

Τα δυσκολότερα μέρη της παραπάνω διαδικασίας είναι η αναζήτηση του συμπλόκου, που περιέχει την παραληφθείσα λέξη w και στη συνέχεια η αναζήτηση μιας λέξης με το μικρότερο βάρος σ' αυτό το σύμπλοκο. Θα χρησιμοποιήσουμε τον parity-check πίνακα για να αναπτύξουμε μία στρωτή διαδικασία ώστε να απλοποιήσουμε αυτές τις εργασίες.

Έστω C ένας γραμμικός κώδικας μήκους n και διάστασης k. Έστω H ο parity-check πίνακας για τον C. Για κάθε λέξη w του K^n, ως *σύνδρομο* της w ορίζουμε τη λέξη wH του K^{n-k}.

Παράδειγμα 2.11.3 Για τον κώδικα C του παραδείγματος 2.11.1 που είδαμε παραπάνω, ο παρακάτω πίνακας H είναι ένας parity-check πίνακας. Εάν $w = 1101$, τότε το σύνδρομο της w είναι η

$$wH = 1101 \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = 11.$$

Ας παρατηρήσουμε ότι η λέξη με το ελάχιστο βάρος στο σύμπλοκο $C + w$ είναι η $u = 1000$ (δείτε παράδειγμα 2.11.1) και το σύνδρομο της u είναι η

$$uH = 1000 \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix} = 11 = wH.$$

Επίσης, εάν $w = 1101$ είναι η λέξη που παραλαμβάνεται, η ΠΑΜΠ θα συμπεράνει ότι η $v = w + u = 1101 + 1000 = 0101$ μας στάλθηκε, οπότε εμφανίζεται ένα σφάλμα στο πρώτο ψηφίο. Ας παρατηρήσουμε επίσης ότι για το υπόδειγμα λάθους της u, το σύνδρομο uH επιλέγει εκείνη τη γραμμή του H, δηλαδή την πρώτη που αντιστοιχεί στη θέση με το πιο πιθανό λάθος.

Το παρακάτω θεώρημα περιέχει κάποιες βασικές και χρήσιμες προτάσεις για τις σύνδρομες λέξεις. Οι αποδείξεις μπορούν να κατασκευαστούν χρησιμοποιώντας τους ορισμούς των εμπλεκόμενων εννοιών και τις ιδιότητες των συμπλόκων από το θεώρημα 2.10.3.

Θεώρημα 2.11.4 *Έστω C είναι ένας γραμμικός κώδικας μήκους n. Έστω H είναι ο parity-check πίνακας για τον C. Έστω w και u δύο λέξεις του k^n.*

1. *$wH = 0$ αν και μόνο αν η w είναι μία κωδικολέξη του C.*
2. *$wH = uH$ αν και μόνο αν οι w και u βρίσκονται στο ίδιο σύμπλοκο του C.*
3. *Εάν η u είναι ένα υπόδειγμα λάθους σε μια παραληφθείσα λέξη w, τότε η uH είναι το άθροισμα των γραμμών του H, που αντιστοιχούν σε εκείνες τις θέσεις στις οποίες εμφανίστηκαν λάθη κατά τη μετάδοση του μηνύματος.*

Ας παρατηρήσουμε ότι εάν δεν εμφανίζονται λάθη κατά τη μετάδοση και η w παραλαμβάνεται, τότε $wH = 0$. Όμως το $wH = 0$ δε συνεπάγεται ότι δεν εμφανίστηκαν λάθη, επειδή η κωδικολέξη w, δεν είναι κατ' ανάγκη εκείνη, η οποία μας αποστάλθηκε.

Επειδή λέξεις του ίδιου συμπλόκου έχουν τις ίδιες συνδρόμους και λέξεις διαφορετικών συμπλόκων έχουν διαφορετικές συνδρόμους, μπορούμε να ταυτίσουμε

ένα σύμπλοκο με τη σύνδρομό του λέξη, όπου το σύνδρομο του συμπλόκου είναι το σύνδρομο μιας οποιοσδήποτε λέξεως που ανήκει στο σύμπλοκο. Έτσι αν ο κώδικας έχει μήκος n και διάσταση k τότε οι 2^{n-k} λέξεις μήκους $n-k$ εμφανίζονται ως οι σύνδρομες ενός και μόνο συμπλόκου από τα 2^{n-k} σύμπλοκα.

Παράδειγμα 2.11.5 Ο κώδικας C, του παραδείγματος 2.11.1 έχει μήκος $n=4$ και διάσταση $k=2$. Τα σύμπλοκα του C (τα οποία καταγράφονται στο παράδειγμα 2.11.1) περιέχουν όλες τις $2^n = 2^4 = 16$ λέξεις μήκους $n=4$. Υπάρχουν $2^{n-k} = 2^{4-2} = 2^2 = 4$ λέξεις μήκους $n-k=2\cdot$ κάθε μία είναι το σύνδρομο ακριβώς ενός από τα $2^{n-k} = 4$ σύμπλοκα του C.

Για να υπολογιστεί το σύνδρομο για ένα σύμπλοκο, μπορούμε να επιλέξουμε μια τυχούσα λέξη w στο σύμπλοκο. Τότε η wH θα είναι το σύνδρομο του συμπλόκου. Για την ΑΜΠ, ζητάμε μια λέξη ελαχίστου βάρους σ' αυτό το σύμπλοκο για να τη χρησιμοποιήσουμε ως υπόδειγμα λάθους. Στα παραδείγματα της τελευταίας ενότητας, διατάξαμε προσεκτικά τα σύμπλοκα ώστε η λέξη με ελάχιστο βάρος να βρίσκεται στην κορυφή ή να αναγράφεται πρώτη. Κάθε λέξη ελαχίστου βάρους σ' ένα σύμπλοκο ονομάζεται *οδηγός του συμπλόκου*. Εάν θα υπάρξουν πάνω από δύο υποψήφιες οδηγοί του συμπλόκου, τότε επιλέγουμε αυθαίρετα τη μία όταν δουλεύουμε με τη ΠΑΜΠ.

Παράδειγμα 2.11.6 Έστω C ξανά να είναι ο κώδικας του παραδειγμάτος 2.11.1. Για κάθε σύμπλοκο υπολογίζουμε τη σύνδρομό του, χρησιμοποιώντας τον οδηγό του συμπλόκου και θέτοντας τα αποτελέσματα στον επόμενο πίνακα.

Οδηγός του συμπλόκου u	Σύνδρομο uH
0000	00
1000	11
0100	01
0010	10

Σημειώνουμε ξανά ότι κάθε λέξη μήκους 2 εμφανίζεται μία και μοναδική φορά ως κάποιο σύνδρομο.

Ο πίνακας του παραπάνω παραδείγματος 2.11.6 που ταιριάζει κάθε σύνδρομο με την οδηγό του συνδρόμου, ονομάζεται *κανονική παράταξη αποκωδικοποίησης* (standard decoding array), εν συντομία ΚΠΑ. Για να κατασκευάσουμε μια ΚΠΑ, πρώτα καταγράφουμε όλα τα σύμπλοκα του κώδικα και επιλέγουμε από κάθε σύμπλοκο μία λέξη ελαχίστου βάρους ως οδηγό του συμπλόκου u. Στη συνέχεια εφευρίσκουμε έναν parity check πίνακα για τον κώδικα και για κάθε οδηγό συνδρόμου u και υπολογίζουμε το σύνδρομο της uH. Ένας πιο γρήγορος τρόπος να κατασκευάσουμε μια ΚΠΑ, δοθέντος του parity check πίνακα H και της απόστασης d του κώδικα C θα είναι να φτιάξουμε όλα τα υποδείγματα λάθους e με $wt(e) \leq \lfloor (d-1)/2 \rfloor$) και να υπολογίσουμε το σύνδρομο $s = eH$ για κάθε μία.

Παράδειγμα 2.11.7 Θα κατασκευάσουμε μια ΚΠΑ για τον κώδικα C του παραδείγματος 2.10.5 (όπου τα σύμπλοκα του C έχουν ήδη καταγραφεί). Για κάθε ένα από τα πέντε πρώτα σύμπλοκα δεν έχουμε άλλη επιλογή για τον οδηγό του

συμπλόκου, η λέξη στην κορυφή είναι η μοναδική λέξη ελαχίστου βάρους σε κάθε σύμπλοκο. Όμως στο τελευταίο σύμπλοκο, το ελάχιστο βάρος μιας λέξεως είναι 2 και το σύμπλοκο περιέχει τρεις λέξεις με βάρος 2 τις 000101, 001010 και 110000. Χρησιμποιώντας τη *ΠΑΜΠ* θα μπορούσαμε να επιλέξουμε την 000101 ως το υποτιθέμενο υπόδειγμα λάθους. Χρησιμοποιώντας την *ΗΑΜΠ* θα ζητήσουμε επαναμετάδοση και θα θέσουμε ένα «*» σε κάθε μία από τις αναφερθείσες θέσεις της ΚΠΑ. Μπορούμε να φτιάξουμε τον ακόλουθο parity check πίνακα για τον C:

$$H = \begin{bmatrix} 110 \\ 011 \\ 111 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Θα μπορούσαμε να κατασκευάσουμε την παρακάτω ΚΠΑ για τον C, υποθέτοντας ότι η ΠΑΜΠ χρησιμποιείται:

Υπόδειγμα Λάθους	Σύνδρομη Λέξη uH
000000	000
100000	110
010000	011
001000	111
000100	100
000010	010
000001	001
000101	101

Ας παρατηρήσουμε ότι οι σύνδρομες λέξεις είναι όλες οι λέξεις του K^3. Το σύμπλοκο C έχει πάντοτε τη μηδενική λέξη ως την οδηγό του και πάντοτε έχει ως σύνδρομο τη μηδενική. Εάν επιλέξουμε για το τελευταίο σύμπλοκο ως οδηγό τη $u = 000101$, παίρνουμε ως σύνδρομο τη $uH = 101$, που είναι το άθροισμα των γραμμών 4 και 6 του H, δηλαδή οι θέσεις που βρίσκονται οι μονάδες του υποδείγματος λάθους u. Χρησιμποιώντας την ΗΑΜΠ, αυτή ακριβώς η θέση θα πάρει «*».

Ασκήσεις

2.11.8 Κατασκευάστε μια ΚΠΑ υποθέτοντας την ΗΑΜΠ για κάθε έναν από τους κώδικες της άσκησης 2.10.6.

2.11.9 Κατασκευάστε μια ΚΠΑ υποθέτοντας την ΗΑΜΠ για κάθε έναν από τους κώδικες της άσκησης 2.10.7.

2.11.10 Κατασκευάστε μια ΚΠΑ υποθέτοντας την ΗΑΜΠ για κάθε έναν από τους κώδικες της άσκησης 2.10.8.

2.11.11 Αποδείξτε το θεώρημα 2.11.4.

Τελικά μπορούμε να κάνουμε κάποια αποκωδικοποίηση. Εάν τα καταφέρουμε στο δύσκολο έργο της κατασκευής μιας ΚΠΑ, τότε είναι εύκολο να χρησιμοποιήσουμε την ΑΜΠ. Έτσι, όταν λαμβάνουμε μία λέξη w, υπολογίζουμε πρώτα το σύνδρομο wH. Στη συνέχεια βρίσκουμε την οδηγό του συμπλόκου u, που βρίσκεται δίπλα στο σύνδρομο $wH = uH$ της ΚΠΑ. Συμπαιρένουμε ότι η $v = w + u$ μας αποστάλθηκε με τη μεγαλύτερη πιθανότητα.

Παράδειγμα 2.11.12 Έστω C είναι ο κώδικας του παραδείγματος 2.11.1. Μια ΚΠΑ βρίσκεται στο παράδειγμα 2.11.6. Ο parity-check πίνακας H βρίσκεται στο παράδειγμα 2.11.3. Υποθέτουμε ότι η $w = 1101$ λαμβάνεται. Τότε το σύνδρομο είναι $wH = 11$, που βρίκεται στη $2\overset{\eta}{=}$ γραμμή της ΚΠΑ, όπου η οδηγός του συμπλόκου είναι η $u = 1000$. Συμπαιρένουμε ότι η $u = w + u = 0101$ έχει αποσταλεί. Εάν η $w = 1111$ έχει ληφθεί, τότε $wH = 01 = uH$, όπου $u = 0100$ από την ΚΠΑ. Αποκωδικοποιούμε την w ως $u = w + u = 1011$. Αυτά τα αποτελέσματα είναι τα ίδια όπως στο παράδειγμα 2.11.1.

Εάν η $w = 1101$ λαμβάν decoded $v = 0101$ εται, τότε αποκωδικοποιούμε $u = 0101$ ως η λέξη που στάλθηκε. Οι παρακάτω υπολογισμοί:

$$d(0000, 1101) = 3 \qquad d(0101, 1101) = 1$$

$$d(1011, 1101) = 2 \qquad d(1110, 1101) = 2$$

μας δίνουν τις αποστάσεις μεταξύ του w και κάθε κωδικολέξης του C και δείχνουν ότι πράγματι η $u = 0101$ είναι η κοντινότερη λέξη του C στο w.

Για $w = 1111$ να λαμβάνεται, οι ίδιοι υπολογισμοί:

$$d(0000, 1111) = 4 \qquad d(0101, 1111) = 2$$

$$d(1011, 1111) = 1 \qquad d(1110, 1111) = 1$$

αποκαλύπτουν ότι υπάρχει ένα μπέρδεμα για την κοντινότερη λέξη του C στην w. Αυτό ήταν αναμενόμενο διότι υπήρχε μία επιλογή για την οδηγό του συμπλόκου που περιείχε την w. Δουλεύουμε με ΠΑΜΠ, οπότε επιλέξαμε αυθαίρετα μία οδηγό του συμπλόκου και κατά συνέπεια επιλέξαμε αυθαίρετα μία λέξη του C κοντινότερα στην w.

Παράδειγμα 2.11.13 Έστω C είναι ο κώδικας του παραδείγματος 2.10.5. Μια ΚΠΑ κατασκευάστηκε στο παράδειγμα 2.11.7. Κάνουμε κάποια αποκωδικοποίηση χρησιμοποιώντας αυτήν την ΚΠΑ. Υποθέστε ότι λαμβάνουμε $w = 110111$. Τότε $wH = 010$, η οποία βρίσκεται στην $6\overset{\eta}{=}$ γραμμή της ΚΠΑ. Η οδηγός συμπλόκου σ' αυτήν τη γραμμή είναι η $u = 000010$. Οπότε η ΠΑΜΠ συμπεραίνει ότι η $v = w + u = 110111 + 000010 = 110101$ είναι η κωδικολέξη που στάλθηκε. Ας υποθέσουμε τώρα ότι η $w = 110000$ λαμβάνεται. Το σύνδρομο $wH = 101$ μας οδηγεί στην τελευταία γραμμή της ΚΠΑ, όπου η οδηγός συμπλόκου είναι η $u = 000101$. Αποκωδικοποιούμε την w ως $v = w+u = 110000+000101 = 110101$. Εάν είχαμε όμως επιλέξει την $u' = 001010$ ως οδηγό συμπλόκου για το τελευταίο σύμπλοκο, τότε θα μπορούσαμε να αποκωδικοποιήσουμε την w ως $w + u' = 110000 + 001010 = 111010$.

Ασκήσεις

2.11.14 Συνεχίζοντας το τελευταίο παράδειγμα με την $w = 110000$ να λαμβάνεται. Αποκωδικοποιείστε υποθέτοντας ότι η $u'' = 110000$ έχει επιλεγεί ως οδηγός συμπλόκου για το τελευταίο σύμπλοκο.

2.11.15 Ας υποθέσουμε ότι η $w = 110111$ λαμβάνεται στο παράδειγμα 2.11.13. Ελέγξτε ότι πράγματι η $u = 110101$ είναι η κοντινότερη κωδικολέξη του C στην w.

2.11.16 Ξανά στο παράδειγμα 2.11.13 με την $w = 110000$ να λαμβάνεται. Βρείτε όλες τις κωδικολέξεις του C κοντινότερα στην W.

2.11.17 Επαναλάβετε την αποκωδικοποίηση στην άσκηση 2.11.2 χρησιμοποιώντας την ΚΠΑ στο παράδειγμα 2.11.7.

2.11.18 Για τον κώδικα του παραδείγματος 2.11.13 παραπάνω, αποκωδικοποιείστε τις παρακάτω παραληφθείσες λέξεις w.

(α') 011101 (β') 110101 (γ') 111111 (δ')000000.

2.11.19 Για κάθε έναν από τους παρακάτω κώδικες, χρησιμοποιείστε την ΚΠΑ για να αποκωδικοποιήσετε τις παρακάτω ληφθείσες λέξεις (οι ΚΠΑ για αυτούς τους κώδικες κατασκευάστηκαν στις ασκήσεις 2.11.8 και 2.11.9).

(α'). $C = \{0000, 1001, 0101, 1100\}$

(i) $w = 1110$ (ii) $w = 1001$ (iii) $w = 0101$

(β'). $C = \{00000, 10100, 01011, 11111\}$

(i) $w = 10101$ (ii) $w = 01110$ (iii) $w = 10001$

(γ'). $C =< 111000, 001110, 100011 >$

(i) $w = 101010$ (ii) $w = 011110$ (iii) $w = 011001$

2.11.20 Έστω C είναι ο κώδικας με parity-check πίνακα

$$H = \begin{bmatrix} 011 \\ 101 \\ 110 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Αποκωδικοποιείστε

(α'). 110100

(β΄). 111111

(γ΄). 101010

(δ΄). 000110.

2.11.21 Έστω C είναι ο κώδικας μήκους 7, ο οποίος έχει έναν parity-check πίνακα H διαστάσεων 7×3, οι γραμμές του οποίου είναι όλες μη μηδενικές λέξεις μήκους 3.

(α΄). Κατασκευάστε μια ΚΠΑ για τον C.

(β΄). Αποκωδικοποιείστε την 1010101.

Εάν θέλαμε να κατασκευάσουμε μια ΚΠΑ, όταν χρησιμοποιούμε την ΗΑΜΠ, μπορούμε να δουλέψουμε ως εξής: Εάν η λέξη w έχει παραληφθεί, τότε το πλήθος των λέξεων του C κοντινότερα στην w είναι το ίδιο με το πλήθος των υποδειγμάτων λάθους στο σύμπλοκο $C + w$, που έχουν ελάχιστο βάρος. Εάν σε κάποιο σύμπλοκο του C υπάρχουν παραπάνω από μία λέξεις ελαχίστου βάρους, τότε αυτό το σύμπλοκο και το σύνδρομό του απαλείφονται από την ΚΠΑ, όταν χρησιμποιούμε την ΗΑΜΠ. Επίσης, το βάρος της οδηγού συμπλόκου είναι το πλήθος των λαθών που διορθώθηκαν από την ΗΑΜΠ, όταν η μία λέξη σ' αυτό το σύμπλοκο έχει παραληφθεί. Εάν αυτό το βάρος είναι υπερβολικά υψηλό, τότε θα μπορούσαμε να απαλείψουμε αυτό το σύμπλοκο και το σύνδρομό του από την ΚΠΑ για την ΗΑΜΠ, ακόμα και αν υπάρχει μόνο μία λέξη ελαχίστου βάρους σ' αυτό το σύμπλοκο. Για να μπορέσουμε να χρησιμοποιήσουμε την κολοβή ΚΠΑ για την ΗΑΜΠ, εάν η ληφθείσα λέξη έχει σύνδρομο που δεν εμφανίζεται στην ΚΠΑ, ζητάμε αναμετάδοση.

Στην πράξη δεν είναι ασυνήθιστο να έχουμε μία τάξη 2^{50}, περίπου 1.126×10^{15} οδηγούς συμπλόκων και σύνδρομα κάτι που κάνει την ΚΠΑ για έναν τυχαίο γραμμικό κώδικα τρομερά δύσχρηστη. Έτσι στην πράξη, δεν έχουμε λύσει το πρόβλημα της αποκωδικοποίησης χρησιμοποιώντας την ΑΜΠ. Όπως θα δούμε αργότερα, η ΑΜΠ είναι υπολογιστικά εφικτή εάν ο γραμμικός κώδικας κατασκευάζεται με ορισμένες προδιαγραφές. Πράγματι, ένας σκοπός της θεωρίας κωδίκων είναι να κατασκευάσει κώδικες που είναι εύκολο να αποκωδικοποιηθούν χρησιμοποιώντας την ΑΜΠ.

2.12 Αξιοπιστία της ΗΑΜΠ για γραμμικούς κώδικες

Έστω C είναι ένας γραμμικός κώδικας μήκους n και διάστασης k. Ας θυμηθούμε ότι η $\theta_p(C, v)$ είναι η πιθανότητα που εάν η v του C αποστέλλεται ενός ΔΣΚ με πιθανότητα (αξιοπιστία) p, τότε η ΗΑΜΠ θα συμπεράνει σωστά ότι η v αποστάλθηκε.

Για κάθε έναν οδηγό συμπλόκου u και για κάθε κωδικολέξη v του C η $v + u$ είναι κοντινότερα στην v παρά σε οποιαδήποτε άλλη κωδικολέξη. Επίσης αν $w \neq$

$v + u$ για κάποια κωδικολέξη v και κάποιο οδηγό συμπλόκου u τότε η w είναι τόσο κοντά στην v όσο κοντά είναι σε κάποια άλλη κωδικολέξη. Οπότε για ένα γραμμικό κώδικα, το σύνολο $L(v)$ των λέξεων που βρίσκονται κοντινότερα στην v, παρά σε οποιαδήποτε άλλη κωδικολέξη είναι:

$$L(v) = \{w | w = v + u \text{ όπου } u \text{ είναι μία μοναδική οδηγός συμπλόκου.}\}$$

Εάν $w = v + u$ τότε η $\theta_p(v, w)$ εξαρτάται μόνο από το $wt(u)$· έτσι για ένα γραμμικό κώδικα C, η $\theta_p(C, v)$ δεν εξαρτάται από την v. Συμβολίζουμε αυτή την κοινή τιμή με $\theta_p(C)$ και άρα

$$\theta_p(C) = \sum_{u \in L(0)} p^{n-wt(u)}(1-p)^{wt(u)}.$$

Συνεπώς, για να βρούμε την αξιοπιστία ενός γραμμικού κώδικα χρειάζεται να επικεντρωθούμε μονάχα στους μοναδικούς οδηγούς συμπλόκων. Απλώς υπολογίζουμε την πιθανότητα κάθε μοναδικού οδηγού συμπλόκου, που εμφανίζεται ως υπόδειγμα λάθους και αφού αθροίσουμε αυτές τις πιθανότητες παίρνουμε την $\theta_p(C)$.

Σημειώστε ότι έχουμε επίσης αποδείξει ότι για ένα γραμμικό κώδικα, το σύνολο των υποδειγμάτων λάθους τα οποία μπορούν να διορθωθούν χρησιμοποιώντας την ΗΑΜΠ είναι ίσο με το σύνολο των μοναδικών οδηγών συμπλόκων.

Παράδειγμα 2.12.1 Έστω C είναι ο κώδικας του παραδείγματος 2.10.5. Χρησιμοποιώντας την ΗΑΜΠ βρίσκουμε ότι υπάρχει μία οδηγός συμπλόκου βάρους 0 και έξι βάρους 1. Άρα

$$\theta_p(C) = p^6 + 6p^5(1-p).$$

Ασκήσεις

2.12.2 Υπολογίστε την $\theta_p(C)$ για καθέναν από τους κώδικες στις ασκήσεις 2.10.6, 2.10.7 και 2.10.8.

Κεφάλαιο 3

Τέλειοι και σχετικοί κώδικες

3.1 Μερικά φράγματα για κώδικες

Στρέφουμε την προσοχή μας στο πρόβλημα να βρούμε πόσες λέξεις ένας γραμμικός κώδικας δοσμένου μήκους h και απόστασης d μπορεί να έχει. Το πρόβλημα στη γενική περίπτωση είναι ανοικτό, αν και έχει απαντηθεί για κάποιες τιμές του n και του d. Όμως μπορούμε να βρούμε κάποια φράγματα στο μέγεθος του κώδικα με δοθέντες παραμέτρους.

Υπενθυμίζουμε ότι εάν t και n είναι ακέραιοι, $0 \leq t \leq n$, τότε το σύμβολο

$$\binom{n}{t} = \frac{n!}{t!(n-t)!},$$

είναι το πλήθος των διαφορετικών τρόπων που μία -μη διατεταγμένη- συλλογή από t αντικείμενα μπορεί να επιλεγεί από ένα σύνολο από n αντικείμενα. *Έτσι* $\binom{n}{t}$ *είναι το πλήθος των λέξεων μήκους* n *και βάρους* t.

Θεώρημα 3.1.1 *Εάν* $0 \leq t \leq n$ *και αν* v *μία λέξη μήκους* n *τότε το πλήθος των λέξεων μήκους* n *και απόστασης από το* v *το πολύ* t *είναι ακριβώς*

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + ... + \binom{n}{t}.$$

Επειδή υπάρχουν 2^n λέξεις μήκους n, θέτοντας $t = n$ στο θεώρημα 3.1.1 παίρνουμε:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + ... + \binom{n}{n} = 2^n.$$

Ασκήσεις

3.1.2 Δώστε ένα παράδειγμα του θεωρήματος 3.1.1 παίρνοντας $v = 10110$ και $t = 3$ και βρίσκοντας όλες τις λέξεις του K^5 με απόσταση το πολύ 3 από την v και στη συνέχεια ελέγχοντας ότι πράγματι το θεώρημα 3.1.1 δίνει το σωστό αριθμό τέτοιων λέξεων.

Για να βρούμε όλες τις λέξεις μιας δοθείσας απόστασης t από μια σταθερή λέξη v, απλά προσθέτουμε στην v, όλες τις λέξεις με βάρος t. Υπάρχουν $\binom{n}{t}$ τέτοιες λέξεις. Εάν ο C είναι ένας κώδικας με μήκος n και απόσταση $d = 2t + 1$ τότε δεν υπάρχει λέξη w με απόσταση το πολύ t από δύο διαφορετικές κωδικολέξεις v_1 και v_2. Πράγματι, εάν $d = (w, v_1) \leq t$ και $d = (w, v_2) \leq t$ με $v_1 \neq v_2$, τότε:

$$d(v_1, v_2) \leq d(v_1, w) + d(w, v_2) \leq 2t < d = 2t + 1,$$

το οποίο είναι αδύνατο διότι ο C έχει ελάχιστη απόσταση d. Έτσι εάν ο C έχει μήκος n και απόσταση $2t + 1$, τότε η λίστα των λέξεων του K^n με απόσταση το πολύ t από μία κωδικολέξη v_1 δεν έχει κοινή κωδικολέξη με τη λίστα των λέξεων με απόσταση το πολύ t από μία άλλη κωδικολέξη v_2, $v_1 \neq v_2$. Αυτό μας δίνει το παρακάτω αποτέλεσμα.

Θεώρημα 3.1.3 *(Φράγμα Hamming). Εάν ο C είναι ένας κώδικας μήκους n και απόστασης $d = 2t + 1$ ή $2t + 2$ τότε:*

$$|C|\left(\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t}\right) \leq 2^n,$$

ή

$$|C| \leq \frac{2^n}{\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t}}.$$

Το φράγμα του Hamming είναι άνω φράγμα για το πλήθος των λέξεων σε έναν κώδικα (γραμμικό ή όχι) μήκους n και απόστασης $d = 2t+1$. Σημειώστε ότι επειδή το $t = \lfloor (d-1)/2 \rfloor$, έχουμε από το θεώρημα 1.12.9, ότι ένας τέτοιος κώδικας θα διορθώνει υποδείγματα λάθους με βάρος μικρότερο ή ίσο του t.

Παράδειγμα 3.1.4 Υπολογίζουμε ένα άνω φράγμα για το μέγεθος ή αλλιώς τη διάσταση k ενός γραμμικού κώδικα με μήκος $n = 6$ και απόσταση $d = 3$. Από $d = 3 = 2t + 1$ παίρνουμε $t = 1$. Το φράγμα Hamming δίνει:

$$|C| \leq \frac{2^6}{\binom{6}{0} + \binom{6}{1}} = \frac{64}{1+6} = \frac{64}{7}.$$

Όμως ο $|C|$ πρέπει να είναι μία δύναμη του 2, οπότε $|C| \leq 8$ και άρα $k \leq 3$.

Ασκήσεις

3.1.5 Βρείτε ένα πάνω φράγμα για το μέγεθος ή τη διάσταση ενός γραμμικού κώδικα με τις παρακάτω τιμές των n και d.

(α'). $n = 8$, $d = 3$

(β'). $n = 7$, $d = 3$

(γ'). $n = 10$, $d = 5$

(δ'). $n = 15$, $d = 3$

(ε'). $n = 15$, $d = 5$

(ϛ'). $n = 23$, $d = 7$

3.1.6 Βεβαιώστε το φράγμα Hamming για το γραμμικό κώδικα C με γεννήτορα πίνακα τον δοθέντα.

(α'). $G = \begin{bmatrix} 111110000000000 \\ 000001111100000 \\ 000001111111111 \end{bmatrix}$

(β'). $G = \begin{bmatrix} 100111 \\ 010101 \\ 001011 \end{bmatrix}$

(γ'). $G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}$.

Το παρακάτω άνω φράγμα λέγεται φράγμα Singleton:

Θεώρημα 3.1.7 *Για κάθε (n, k, d) γραμμικό κώδικα, $d - 1 \leq n - k$.*

Απόδειξη: Ας θυμηθούμε ότι από την παράγραφο 2.7 και το θεώρημα 2.9.1, γνωρίζουμε ότι ο parity check πίνακας H ενός (n, k, d) γραμμικού κώδικα είναι ένας $n \times (n - k)$ πίνακας τέτοιος ώστε κάθε $d - 1$ γραμμές του H είναι γραμμικά ανεξάρτητες. Επειδή οι γραμμές έχουν μήκος $n - k$, δεν μπορούμε να έχουμε περισσότερα από $n - k$ ανεξάρτητα διανύσματα γραμμές. Οπότε $d - 1 \leq n - k$ ή ισοδύναμα $k \leq n - d + 1$.

Το φράγμα Singleton (θεώρημα 3.1.7) είναι κάποια έννοια ασθενέστερη από το φράγμα του Hamming. Για παράδειγμα, εάν $n = 15$ και $d = 5$ τότε το θεώρημα 3.1.7 δίνει ότι $k \leq 11$, ενώ το θεώρημα 3.1.3 (φράγμα Hamming) δίνει $k \leq 9$. Εντούτοις μερικοί κώδικες πράγματι δίνουν την ισότητα στο φράγμα Singleton οπότε το φράγμα Singleton χρησιμοποιείται για να ορίσουμε μια σπουδαία και χρήσιμη κλάση κωδίκων που λέγονται διαχωρίσιμοι κώδικες μεγίστης απόστασης.

Ένας γραμμικός (n, k, d) κώδικας ονομάζεται *διαχωρίσιμος κώδικας μεγίστης απόστασης - Maximum Distance Separable* (ή εν συντομία MDS), εάν $d = n-k+1$ (ή $k = n - d + 1$). Υπάρχουν πολλοί χαρακτηρισμοί των κωδίκων.

Θεώρημα 3.1.8 *Για ένα (n, k, d) γραμμικό κώδικα C, τα παρακάτω είναι ισοδύναμα:*

(1) $d = n - k + 1$,

(2) κάθε $n - k$ γραμμές του parity check πίνακα είναι γραμμικά ανεξάρτητες,

(3) κάθε k στήλες του γεννήτορα πίνακα είναι γραμμικά ανεξάρτητες και

(4) ο C είναι MDS.

Απόδειξη: Το θεώρημα 3.1.7 δίνει $d \leq n-k+1$· όμως $d \geq n-k+1$ εάν κάθε $n-k$ γραμμές του parity check πίνακα είναι ανεξάρτητες. Έτσι τα (1) και (2) είναι ισοδύναμα. Για το (3) παρατηρείστε ότι εάν $d = n-k+1$, καμία μη μηδενική κωδικολέξη δεν μπορεί να έχει περισσότερα από $k-1$ μηδενικά. Όμως, k στήλες του $k \times n$ γεννήτορα πίνακα είναι γραμμικά ανεξάρτητες εάν κάποια μη μηδενική κωδικολέξη έχει k μηδενικά σε αυτές ισόβαθμες τις στήλες. Αυτό είναι σχετικά εύκολο να το δούμε και το αφήνουμε για τις ασκήσεις.

Πόρισμα 3.1.9 *Ο δυϊκός κώδικας ενός $(n, k, n-k+1)$ MDS κώδικα είναι ένας $(n, n-k, k+1)$ MDS κώδικας.*

Θα ξανασυζητήσουμε αργότερα για MDS κώδικες όταν θα μελετούμε τους Reed-Solomon κώδικες.

Ασκήσεις

3.1.10 Οι στήλες 2, 3 και 5 του γεννήτορα πίνακα G παρακάτω είναι γραμμικά ανεξάρτητες. Βρείτε μια κωδικολέξη που έχει μηδενικά στις θέσεις 2, 3 και 5

$$G = \begin{bmatrix} 11001 \\ 01110 \\ 00101 \end{bmatrix}.$$

3.1.11 Δείξτε ότι εάν ένας $k \times n$ γεννήτορας πίνακας έχει k γραμμικά εξαρτημένες στήλες τότε υπάρχει μία μη μηδενική κωδικολέξη με μηδενικά σε αυτές τις k θέσεις.

Ακόμα δεν έχουμε κατασκευάσει κώδικες για δοθέντες παραμέτρους n, k και d. Τα άνω όρια αποκλείουν κάποιες τιμές, για παράδειγμα το φράγμα Hamming μας λέει ότι ένας κώδικας με μήκος $n = 15$ και απόσταση $d = 5$ δεν μπορεί να έχει διάσταση $k = 10$. Εντούτοις αυτό το φράγμα δεν αποκλείει την πιθανότητα να υπάρχει ένας $(15, 8, 5)$ κώδικας.

Πώς θα μπορούσαμε να βρούμε έναν $(15, 8, 5)$ κώδικα; Γενικά αυτό είναι ένα πολύ δύσκολο πρόβλημα. Μία προσέγγιση θα ήταν να βρούμε τον parity check πίνακα για έναν τέτοιο κώδικα. Δηλαδή, υποθέτοντας $r = n-k$, πρέπει να βρούμε n διανύσματα μήκους r για να σχηματίσουμε τις γραμμές του H έτσι ώστε κάθε σύνολο από $d-1$ τέτοια διανύσματα να είναι γραμμικά ανεξάρτητο.

Παράδειγμα 3.1.12 Έστω $n = 15$, $k = 6$ και $d = 5$. Τότε $r = 15-6 = 9$. Οπότε επιθυμούμε να βρούμε 15 μη μηδενικά διανύσματα μήκους 9 με την ιδιότητα κάθε 4 από αυτά να είναι γραμμικά ανεξάρτητα. Η εύρεση των πρώτων 9 γραμμών είναι εύκολη: πάρτε τον 9×9 ταυτοτικό πίνακα I_9.

Υποθέστε ότι έχουμε με κάποιο τρόπο βρει 3 ακόμα διανύσματα από ένα σύνολο 12 γραμμών και έτσι έχουμε,

$$G = \begin{bmatrix} I_9 \\ 111100000 \\ 100011100 \\ 101000011 \\ ? \end{bmatrix}.$$

Πριν ψάξουμε για το επόμενο διάνυσμα παρατηρούμε ότι το επόμενο λογικό επιχείρημα μας βεβαιώνει ότι τουλάχιστον άλλο ένα υπάρχει. Μεταξύ όλων των 2^9 διανυσμάτων, δεν μπορούμε να επιλέξουμε το μηδενικό ή κάποιο άλλο από τα 12 ήδη επιλεγμένα. Οπότε αποκλείουμε $1 + 12$ διανύσματα. Επίσης αποκλείουμε κάθε διάνυσμα το οποίο μπορεί να γραφτεί ως άθροισμα 2 ή 3 από αυτά τα διανύσματα, διότι αυτό θα δημιουργούσε ένα εξαρτημένο σύνολο από 3 ή 4 διανύσματα αντίστοιχα. Αυτό αποκλείει το πολύ $\binom{12}{2} + \binom{12}{3}$ πρόσθετα διανύσματα. Εντούτοις, οποιοδήποτε διάνυσμα από τα υπόλοιπα μπορεί να επιλεγεί. Επειδή

$$1 + \binom{12}{1} + \binom{12}{2} + \binom{12}{3} < 2^9$$

ξέρουμε ότι μπορούμε να βρούμε ακόμα ένα διάνυσμα. Για παράδειγμα, κάποιος θα μπορούσε να επιλέξει το διάνυσμα 010101010 να είναι η επόμενη γραμμή του H. Η εύρεση και των υπολοίπων γραμμών του H αφήνεται στην άσκηση 3.1.21.

Το παράδειγμα 3.1.12 (και οι σχετιζόμενες ασκήσεις) δείχνουν ότι ένας $(15, 6, 5)$ κώδικας υπάρχει. Αυτό μας δίνει ένα *κάτω φράγμα* στο μέγιστο μέγεθος (ή διάσταση) ενός γραμμικού κώδικα με $n = 15$ και $d = 5$, δηλαδή $6 \leq k \leq 8$.

Το επόμενο αποτέλεσμα μας γενικεύει τη μεθοδολογία του παραδείγματος 3.1.12 για την κατασκευή γραμμικών κωδίκων (και έτσι θέτει κάτω φράγματα). Οι αποδείξεις αφήνονται στην άσκηση 3.1.22.

Θεώρημα 3.1.13 *(Φράγμα Gilbert-Varshamov). Υπάρχει ένας γραμμικός κώδικας με μήκος n, διάσταση k και απόσταση d εάν*

$$\binom{n-1}{0} + \binom{n-1}{1} + ... + \binom{n-1}{d-2} < 2^{n-k}.$$

Πόρισμα 3.1.14 *Εάν $n \neq 1$ και $d \neq 1$ τότε υπάρχει ένας γραμμικός κώδικας C με μήκος n και απόσταση d με*

$$|C| \geq \frac{2^{n-1}}{\binom{n-1}{0} + \binom{n-1}{1} + ... + \binom{n-1}{d-2}}.$$

Παράδειγμα 3.1.15 Υπάρχει ένας γραμμικός κώδικας με μήκος $n = 5$, διάσταση $k = 2$ και απόσταση $d = 5$;

Για να αποφασίσουμε αν ένας τέτοιος κώδικας υπάρχει, βρίσκουμε το Gilbert-Varshamov φράγμα:

$$\begin{aligned}\binom{n-1}{0}+...+\binom{n-1}{d-2} &= \binom{8}{0}+\binom{8}{1}+\binom{8}{2}+\binom{8}{3}\\ &= 93\end{aligned}$$

και $2^{n-k}=2^{9-2}=2^7=128$. Επειδή $93<128$, τέτοιος κώδικας υπάρχει.

Παράδειγμα 3.1.16 Υπάρχει ένα κάτω και ένα άνω φράγμα στο μέγεθος ή τη διάσταση k, ενός κώδικα με $n=9$ και $d=5$;
Για να βρούμε ένα κάτω φράγμα για το μέγιστο πλήθος κωδικολέξεων που ένας τέτοιος κώδικας C θα έχει, χρησιμοποιούμε το πόρισμα 3.1.14:

$$|C|\geq \frac{2^{n-1}}{\binom{n-1}{0}+...+\binom{n-1}{d-2}}=\frac{2^{9-1}}{\binom{8}{0}+\binom{8}{1}+\binom{8}{2}+\binom{8}{3}}=\frac{2^8}{93}=\frac{256}{93}=2.75.$$

Επειδή το $|C|$ είναι δύναμη του 2, $|C|\geq 4$.

Για να βρούμε ένα άνω φράγμα για το $|C|$, χρησιμοποιούμε το φράγμα Hamming:

$$|C|\leq \frac{2^9}{\binom{9}{0}+\binom{9}{1}+\binom{9}{2}}=\frac{512}{1+9+36}=\frac{512}{46}=11.13.$$

Επειδή το $|C|$ είναι δύναμη του 2, $|C|\leq 8$.

Συνδυάζοντας τα δύο φράγματα, ένας γραμμικός κώδικας με παραμέτρους $(9,k,5)$ με 4 κωδικολέξεις υπάρχει, αλλά δεν υπάρχει $(9,k,5)$ γραμμικός κώδικας με περισσότερες από 8 κωδικολέξεις.

Παράδειγμα 3.1.17 Υπάρχει ένας $(15,7,5)$ γραμμικός κώδικας; Ξανά μπορούμε να χρησιμοποιήσουμε το Gilbert-Varshamov φράγμα για να απαντήσουμε στην ερώτηση.

$$\begin{aligned}\binom{n-1}{0}+...+\binom{n-1}{d-2} &= \binom{14}{0}+\binom{14}{1}+\binom{14}{2}+\binom{14}{3}\\ &= 1+14+91+364\\ &= 470\end{aligned}$$

και $2^{n-k}=2^{15-7}=256$. Σε αυτήν την περίπτωση η ανισότητα δεν ικανοποιείται, οπότε το Gilbert-Varshamov φράγμα δε μας λεει εάν τέτοιος κώδικας υπάρχει ή όχι. Πράγματι, όπως θα δούμε αργότερα, αυτοί είναι οι παράμετροι για τον διόρθωσης λαθών BCH 2- κώδικα, οπότε πράγματι ένας τέτοιος κώδικας υπάρχει.

Ασκήσεις

3.1.18 Για κάθε περίπτωση της άσκησης 3.1.5 θέστε $k = 2d$ και αποφασίστε, εάν είναι εφικτό ή όχι, να υπάρχει ένας γραμμικός κώδικας με τις δοθείσες παραμέτρους. Βρείτε ένα κάτω και ένα άνω φράγμα για το μέγιστο πλήθος των κωδικολέξεων που ένας τέτοιος κώδικας μπορεί να έχει, υποθέτοντας ότι το k δεν είναι φραγμένο.

3.1.19 Βρείτε ένα κάτω και ένα άνω φράγμα για το μέγιστο πλήθος των κωδικολέξεων που είναι γραμμικός κώδικας μήκους n και απόστασης d όπου:

(α'). $n = 15$, $d = 5$

(β'). $n = 15$, $d = 3$

(γ'). $n = 11$, $d = 3$

(δ'). $n = 12$, $d = 3$

(ε'). $n = 12$, $d = 4$

(ϛ'). $n = 12$, $d = 5$

3.1.20 Είναι δυνατό να έχουμε ένα γραμμικό κώδικα με παραμέτρους $(8, 3, 5)$;

3.1.21 Βρείτε έναν $(15, 6, 5)$ κώδικα, φτιάχνοντας τον πίνακα parity check. (Δείτε το παράδειγμα 3.1.12, κάθε ένα από τα 3 ζητούμενα διανύσματα πρέπει να έχει βάρος τουλάχιστον 4. Γιατί;)

3.1.22 Έστω H_i ένας οποιοσδήποτε $i \times (n-k)$ πίνακας του οποίου οποιεσδήποτε $d-1$ γραμμές είναι γραμμικά ανεξάρτητες:

(α'). Δείξτε ότι υπάρχουν το πολύ

$$N_i = \binom{i}{0} + \binom{i}{1} + ... + \binom{i}{d-2}$$

λέξεις στο K^{n-k} οι οποίες είναι γραμμικοί συνδυασμοί το πολύ $d-2$ γραμμών του H_i.

(β'). Δείξτε ότι εάν $N_i < 2^{n-k}$, τότε μία γραμμή μπορεί να προστεθεί με τέτοιο τρόπο ώστε καμία $d-1$ γραμμή του πίνακα που προκύπτει, να είναι γραμμικά εξαρτημένη.

(γ'). Αποδείξτε το φράγμα Gilbert-Varshamov.

(δ'). Αποδείξτε το Πόρισμα 3.1.14.

3.2 Τέλειοι Κώδικες

Ένας κώδικας C μήκους n και περιττής απόστασης $d = 2t + 1$ ονομάζεται *τέλειος κώδικας* εάν ο C πετυχαίνει το φράγμα Hamming του θεωρήματος 3.1.3· δηλαδή, εάν:

$$|C| = \frac{2^n}{\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t}}.$$

Δυστυχώς, δεν υπάρχουν πολλοί τέλειοι γραμμικοί κώδικες, αλλά αυτοί που υπάρχουν είναι πολύ χρήσιμοι. Το κύριο πρόβλημα για την εύρεση γραμμικών τέλειων κωδίκων είναι ότι ο αριθμός $\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t}$ πρέπει να είναι δύναμη του 2 (επειδή ο $|C|$ είναι δύναμη του 2).

Παράδειγμα 3.2.1 Έστω $t = 0$. Τότε $\binom{n}{0} = 1 = 2^0$, οπότε $|C| = 2^n/\binom{n}{0} = 2^n$. Ο μοναδικός κώδικας με 2^n κωδικολέξεις μήκους n είναι ο $C = K^n$. Ο K^n είναι ένας τέλειος κώδικας.

Παράδειγμα 3.2.2 Έστω $n = 2t + 1$. Τότε:

$$\binom{n}{n-i} = \frac{n!}{(n-i)!(n-(n-i))!} = \frac{n!}{(n-i)!n!} = \binom{n}{i}.$$

Οπότε,

$$\binom{n}{0} = \binom{n}{n}, \binom{n}{1} = \binom{n}{n-1}, \binom{n}{2} = \binom{n}{n-2}, ...$$

και από $n = 2t + 1$,

$$\binom{n}{t} = \binom{n}{n-t} = \binom{n}{t+1}.$$

Οπότε,

$$\binom{n}{0} + ... + \binom{n}{t} = \frac{1}{2}\left(\binom{n}{0} + ... + \binom{n}{n}\right) = \frac{1}{2} \cdot 2^n = 2^{n-1}.$$

Έτσι,

$$|C| = \frac{2^n}{\binom{n}{0} + ... + \binom{n}{t}} = \frac{2^n}{2^{n-1}} = 2.$$

Έτσι κάθε τέλειος κώδικας μήκους και απόστασης $2t + 1$ έχει ακριβώς δύο κωδικολέξεις. Μεταξύ των γραμμικών κωδίκων υπάρχει μόνο ένας τέτοιος κώδικας, ο κώδικας που αποτελείται από τη μηδενική λέξη και τη λέξη που κάθε ψηφίο είναι 1 και πράγματι αυτός ο κώδικας είναι τέλειος.

Οι κώδικες στα παραδείγματα 3.2.1 και 3.2.2 παρ' όλο που είναι τέλειοι, δεν είναι πολύ ενδιαφέροντες. Ονομάζονται *τετριμμένοι*, τέλειοι κώδικες.

Παράδειγμα 3.2.3 Έστω $n = 7$ και $d = 3$. Τότε $t = 1$ και

$$|C| = \frac{2^7}{\binom{7}{0} + \binom{7}{1}} = \frac{128}{8} = 16 = 2^4.$$

Οπότε, μπορεί να υπάρχει ένας τέλειος γραμμικός κώδικας με $n = 7$ και $d = 3$. Στην επόμενη ενότητα, θα δούμε ότι υπάρχει ένας τέτοιος κώδικας, ο κώδικας Hamming.

Παράδειγμα 3.2.4 Έστω $n = 23$ και $d = 7$. Τότε $t = 3$ και

$$|C| = \frac{2^{23}}{\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}} = \frac{2^{23}}{1 + 23 + 253 + 1771}$$

$$= \frac{2^{23}}{2048} = \frac{2^{23}}{2^{11}} = 2^{12} = 4096.$$

Αυτό δείχνει ότι μπορεί να υπάρχει ένας τέλειος κώδικας με $n = 23$ και $d = 7$. Στην επόμενη παράγραφο θα δούμε ότι ένας τέτοιος κώδικας υπάρχει, δηλαδή ο κώδικας Golay.

Ασκήσεις

3.2.5 Δείξτε αυτό για $n = 2^r - 1$, $\binom{n}{0} + \binom{n}{1} = 2^r$.

3.2.6 Υπάρχουν τέλειοι κώδικες για τις παρακάτω τιμές των n και d:

(α'). $n = 15$, $d = 3$

(β'). $n = 31$, $d = 3$

(γ'). $n = 15$, $d = 5$

Τα δυνατά μήκη και αποστάσεις για ένα τέλειο κώδικα προσδιορίστηκαν από τους Tietäväiren και van Lint το 1963. Η απόδειξη του αποτελέσματος τους είναι εκτός των σκοπών αυτών σημειώσεων.

Θεώρημα 3.2.7 *Εάν ο C είναι ένας μη τετριμμένος τέλειος κώδικας μήκους n και απόστασης $d = 2t + 1$, τότε είτε $n = 23$ και $d = 7$, είτε $n = 2^r - 1$ για κάποιο $r \geq 2$ και $d = 3$.*

Εάν ένας γραμμικός κώδικας μήκους n έχει απόσταση $d = 2t + 1$, τότε από το θεώρημα 1.12.9, ο C θα διορθώνει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο του $t = (d-1)/2$. Έτσι κάθε λέξη μήκους n και βάρους μικρότερου ή ίσου του t είναι ένας οδηγός συμπλόκου. Υπάρχουν ακριβώς $\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t}$ τέτοιες λέξεις. Αλλά αυτό είναι ακριβώς το πλήθος των συμπλόκων εάν ο κώδικας είναι τέλειος. Οπότε έτσι αποδείξαμε άλλο ένα θεώρημα.

Θεώρημα 3.2.8 *Εάν ο C είναι ένας τέλειος κώδικας μήκους n και απόστασης $d = 2t + 1$, τότε ο C θα διορθώνει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο του t και κανένα άλλο υπόδειγμα λάθους.*

Μπορούμε να ερμηνεύσουμε το θεώρημα 3.2.8 λέγοντας ότι κάθε μια από τις 2^n λέξεις του K^n βρίσκεται σε απόσταση t από μία ακριβώς κωδικολέξη. Αυτή η ιδιότητα μας δίνει τη δυνατότητα να απαριθμήσουμε το πλήθος των κωδικολέξεων ελαχίστου ή μηδενικού βάρους σ' έναν τέλειο κώδικα.

Ένας τέλειος κώδικας ο οποίος διορθώνει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο του t ονομάζεται *τέλειος $t-$διόρθωσης λάθους κώδικας*. Από το θεώρημα 3.2.7 οι μοναδικές δυνατές τιμές για το t είναι $t = 1$ και $t = 3$. Μελετάμε την περίπτωση $t = 1$ στην επόμενη ενότητα.

3.3 Κώδικες Hamming

Τελικά ήρθε η στιγμή να σχεδιάσουμε έναν κώδικα. Θεωρούμε μία σπουδαία οικογένεια κωδίκων οι οποίοι είναι εύκολοι στην κωδικοποίηση και αποκωδικοποίηση και οι οποίοι διορθώνουν όλα τα λάθη με βάρος 1.

Ένας κώδικας με μήκος $n = 2^r - 1$, $r \geq 2$, που έχει parity check πίνακα τον H, του οποίου οι γραμμές αποτελούνται από όλα τα μη μηδενικά διανύσματα μήκους r ονομάζεται *κώδικας Hamming* μήκους $2^r - 1$.

Παράδειγμα 3.3.1 Μια δυνατότητα για έναν parity check πίνακα για έναν κώδικα Hamming μήκους $7(r = 3)$ είναι:

$$H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

Από τον αλγόριθμο 2.5.7, ένας γεννήτορας πίνακας για τον κώδικα Hamming μήκους 7 είναι ο:

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}.$$

Έτσι ο κώδικας έχει διάσταση 4 και περιέχει $2^4 = 16$ κωδικολέξεις. Το θεώρημα 2.9.1 μπορεί να χρησιμοποιηθεί για να βρούμε την απόσταση του κώδικα, η οποία είναι ίση με 3. Ο βαθμός πληροφορίας είναι $4/7$. Στην άσκηση 2.6.12, κωδικοποιήσαμε μερικά μηνύματα χρησιμοποιώντας αυτό τον κώδικα. Υπάρχουν και άλλες πιθανότητες για τον parity check πίνακα ενός κώδικα Hamming μήκους 7, αλλά όλες παράγουν ισοδύναμους κώδικες.

Επειδή ο parity check πίνακας H ενός κώδικα Hamming C περιέχει όλες τις r γραμμές βάρους 1, οι r στήλες του H είναι γραμμικά ανεξάρτητες. Έτσι ένας *κώδικας Hamming έχει διάσταση* $2^r - 1 - r$ *και περιέχει* 2^{2^r-1-r} *κωδικολέξεις*.

Καμία γραμμή του H δεν μπορεί να είναι η μηδενική λέξη, οπότε ο C έχει απόσταση τουλάχιστον 2. Δύο γραμμές του H δεν είναι ποτέ ίσες οπότε δύο γραμμές του H δεν είναι ποτέ γραμμικά εξαρτημένες. Οπότε ο C έχει απόσταση τουλάχιστον 3. Επίσης ο H περιέχει τις γραμμές $100...0$, $010...0$ και $110...0$, οι οποίες σχηματίζουν ένα γραμμικά εξαρτημένο σύνολο. Οπότε, από το θεώρημα 2.9.1, ένας *κώδικας Hamming έχει απόσταση* $d = 3$.

Τώρα για $n = 2^r - 1$ και $d = 2t + 1$ (οπότε $t = 1$),

$$\frac{2^n}{\binom{n}{0} + ... + \binom{n}{t}} = \frac{2^n}{\binom{n}{0} + \binom{n}{1}} = \frac{2^{2^r-1}}{1+n} = \frac{2^{2^r-1}}{1+2^r-1} = 2^{2^r-1-r},$$

οπότε όλοι οι κώδικες Hamming είναι τέλειοι κώδικες. Από το θεώρημα 3.2.8, *οι κώδικες Hamming είναι τέλειοι κώδικες διόρθωσης μοναδικού λάθους*.

Είναι τετριμμένο να φτιάξουμε μια ΚΠΑ για έναν κώδικα Hamming. Όλα τα λάθη με βάρος 1 διορθώνονται οπότε όλες οι λέξεις μήκους $2^r - 1$ και βάρους 1 είναι υποδείγματα λάθους που διορθώνονται, οπότε πρέπει να είναι οδηγοί συμπλόκου. Επειδή εάν e είναι ένα υπόδειγμα λάθους τότε το eH αθροίζει τις γραμμές του parity check πίνακα H που αντιστοιχούν σε θέσεις που υπάρχουν λάθη και επειδή ο H έχει $2^r - 1$ γραμμές, έχουμε την παρακάτω ΚΠΑ για έναν κώδικα Hamming:

οδηγός συμπλόκου	σύνδρομο
$000...0$	$00...0$
I_{2^r-1}	H

Παράδειγμα 3.3.2 Για τον κώδικα Hamming στο παράδειγμα 3.3.1 αποκωδικοποιούμε την $w = 1101001$. Το σύνδρομο είναι $wH = 011$, το οποίο είναι η τέταρτη γραμμή του H. Οπότε η οδηγός του συμπλόκου είναι $I_7 : u = 0001000$. Αποκωδικοποιούμε την w ως $w + u = 1100001$.

Ασκήσεις

3.3.3 Βρείτε ένα γεννήτορα πίνακα σε κανονική μορφή για έναν κώδικα Hamming μήκους 15, και στη συνέχεια αποκωδικοποιείστε το μήνυμα 11111100000.

3.3.4 Κατασκευάστε μια ΚΠΑ για έναν κώδικα Hamming μήκους 7 και χρησιμοποιείστε την για να αποκωδικοποιήσετε τις παρακάτω λέξεις:

(α'). 1101011

(β'). 1111111

(γ'). 0011010

(δ'). 0101011

(ε΄). 0100011

(ϛ΄). 0001011

3.3.5 Κατασκευάστε μια ΚΠΑ για έναν κώδικα Hamming μήκους 15 και χρησιμοποιείστε την για να αποκωδικοποιήσετε τις παρακάτω λέξεις:

(α΄). 01010 01010 01000

(β΄). 11110 00101 10110

(γ΄). 11100 01110 00111

(δ΄). 11100 10110 00000

(ε΄). 00011 10100 00110

(ϛ΄). 11001 11001 11000

3.3.6 Δείξτε ότι καθένας από τους παρακάτω είναι ένας parity check πίνακας για έναν κώδικα Hamming μήκους 7 και ότι και οι δύο κώδικες είναι ισοδύναμοι με εκείνον στο παράδειγμα 3.3.1.

$$H' = \begin{bmatrix} 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{bmatrix} \qquad H'' = \begin{bmatrix} 100 \\ 110 \\ 111 \\ 011 \\ 101 \\ 010 \\ 001 \end{bmatrix}.$$

3.3.7 Δείξτε ότι όλοι οι κώδικες Hamming ενός δοθέντος μήκους είναι ισοδύναμοι.

3.3.8 Είναι ο παρακάτω πίνακας ο αντίστροφος ενός parity check πίνακα για έναν κώδικα Hamming μήκους 15;

$$H^T = \begin{bmatrix} 10001 & 10111 & 01000 \\ 11100 & 10001 & 11110 \\ 01011 & 00101 & 11101 \\ 10001 & 01011 & 00111 \end{bmatrix}$$

3.3.9 Δείξτε ότι ο κώδικας Hamming μήκους $2^r - 1$ για $r = 2$ είναι ο τετριμμένος κώδικας.

3.3.10 Χρησιμοποιείστε τον κώδικα Hamming μήκους 7 στο Παράδειγμα 3.3.1 και τα μηνύματα όπως κωδικοποιήθηκαν στην άσκηση 2.6.11. Αποκωδικοποιείστε το παρακάτω μήνυμα που παραλήφθηκε:

$$1010111, 0110111, 1000010, 0010101, 1001011, 0010000, 1111100.$$

3.4 Εκτεταμένοι Κώδικες

Μερικές φορές η αύξηση του μήκους ενός κώδικα με ένα ψηφίο, ή μερικές φορές με κάποια ψηφία, μας οδηγεί σ' ένα νέο κώδικα με αυξημένες δυνατότητες ανίχνευσης ή διόρθωσης λαθών οι οποίες αξίζουν το τίμημα για ένα χαμηλότερο βαθμό πληροφορίας. Μελετάμε μια απλή περίπτωση σε αυτήν την ενότητα.

Έστω C ένας γραμμικός κώδικας μήκους n. Ο κώδικας C^* μήκους $n+1$ που φτιάχνεται από τον C με την προσθήκη ενός έξτρα ψηφίου σε κάθε κωδικολέξη ώστε να πετύχουμε να έχουμε κάθε λέξη του νέου κώδικα με άρτιο βάρος, ονομάζεται ο *εκτεταμένος κώδικας (extended code)* του C.

Στο παράδειγμα 1.3.3 φτιάξαμε τον εκτεταμένο κώδικα του K^2 και ο (η) αναγνώστης (-τρια) έκανε το ίδιο για τον K^3 στην άσκηση 1.3.5.

Εάν ο αρχικός κώδικας C έχει έναν $k \times n$ γεννήτορα πίνακα G, τότε ο εκτεταμένος κώδικας C^* έχει $K \times (n+1)$ γεννήτορα πίνακα

$$H^* = [G, b],$$

όπου η τελευταία στήλη b του G^* έχει τοποθετηθεί έτσι ώστε κάθε γραμμή του G^* να έχει άρτιο βάρος.

Ένας parity check πίνακας για τον G^* μπορεί να κατασκευάζεται από τον G^* χρησιμοποιώντας τον αλγόριθμο 2.5.7. Αλλά υπάρχει και ένας πιο εύκολος τρόπος εάν μας δίνουν έναν parity check πίνακα H για αρχικό κώδικα C. Σ' αυτήν την περίπτωση, ο εκτεταμένος κώδικας C^* έχει parity check πίνακα

$$G^* = \begin{bmatrix} H & j \\ 0 & 1 \end{bmatrix},$$

όπου j είναι $n \times 1$ στήλη από μονάδες. Παρατηρείστε ότι ο H^* είναι ένας $(n+1) \times (n+1-k)$ πίνακας. Αφού ο H έχει τάξη $n-k$, η τελευταία γραμμή του H^* έχει τάξη $n-k+1$. Ακόμα,

$$G^*H^* = [G, b] \begin{bmatrix} H & j \\ 0 & 1 \end{bmatrix} = [GH, Gj + b].$$

Τώρα $GH = 0$ και ο Gj αθροίζει τις μονάδες σε κάθε γραμμή του G. Από τον ορισμό του b, έχουμε $Gj + b = 0$. Οπότε $G^*H^* = 0$. Από το θεώρημα 2.7.6 οι G^* και H^* είναι πράγματι γεννήτορας και parity check πίνακες αντίστοιχα, για τον γραμμικό κώδικα C^*.

Παράδειγμα 3.4.1 Έστω C ο γραμμικός κώδικας με γεννήτορα πίνακα

$$G = \begin{bmatrix} 10010 \\ 01001 \\ 00111 \end{bmatrix}.$$

Οπότε ο

$$G = \begin{bmatrix} 10 \\ 01 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

είναι ο parity check πίνακας για τον C του αλγόριθμου 2.5.7. Οπότε πετυχαίνουμε τους παρακάτω γεννήτορα και parity check πίνακες για τον εκτεταμένο κώδικα:

$$G^* = \begin{bmatrix} 10010|0 \\ 01001|0 \\ 00111|1 \end{bmatrix} \text{ και } H^* = \begin{bmatrix} 10|1 \\ 01|1 \\ 11|1 \\ 10|1 \\ 01|1 \\ --\,-- \\ 00|1 \end{bmatrix}.$$

Εάν v είναι μία λέξη στον αρχικό κώδικα C και εάν V^* είναι η αντίστοιχη λέξη στον εκτεταμένο κώδικα C^* τότε:

$$wt(v^*) = \begin{cases} wt(v) & \text{εάν } wt(v) \text{ είναι άρτιος} \\ wt(v)+1 & \text{εάν } wt(v) \text{ είναι περιττός.} \end{cases}$$

Οπότε εάν η απόσταση d του C είναι περιττή, τότε η απόσταση του C^* είναι $d+1$, αλλά εάν η d είναι άρτια τότε η απόσταση του C^* είναι d. Οπότε ένας εκτεταμένος κώδικας είναι χρήσιμος όταν η d είναι περιττή, οπότε αυτός δε διορθώνει περισσότερα λάθη από τον C αλλά ανιχνεύει ένα ακόμα λάθος. Σημειώστε τότε, ότι δεν έχει νόημα να επεκτείνουμε ένα κώδικα δύο φορές.

Παράδειγμα 3.4.2 Υποθέστε ότι ο C έχει απόσταση $d = 5$. Οπότε ο C^* έχει απόσταση $d^* = 6$. Από το θεώρημα 1.11.14, ο C ανιχνεύει όλα τα μη μηδενικά υποδείγματα λάθους με βάρος μικρότερο ή ίσο του $d-1=4$ και ο C^* ανιχνεύει όλα τα μη μηδενικά υποδείγματα λάθους με βάρος μικρότερο ή ίσο του $d^*-1=5$. Από το θεώρημα 1.12.9, ο C ανιχνεύει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο του $\lfloor (d-1)/2 \rfloor = \lfloor 4/2 \rfloor = 2$ και ο C^* διορθώνει όλα τα υποδείγματα λάθους με βάρος μικρότερο ή ίσο του $\lfloor (d^*-1)/2 \rfloor = \lfloor 5/2 \rfloor = 2$.

Ασκήσεις

3.4.3 Βρείτε τους γεννήτορα και parity check πίνακες για έναν εκτεταμένο κώδικα Hamming μήκους 8.

3.4.4 Κατασκευάστε μια ΚΠΑ για έναν εκτεταμένο κώδικα Hamming μήκους 8 και χρησιμοποιείστε τον για να αποκωδικοποιήσετε τις παρακάτω λέξεις:

(α'). 10101010

(β'). 11010110

(γ'). 11111111.

3.4.5 Δείξτε ότι ένας εκτεταμένος κώδικας Hamming μήκους 8 είναι ένας αυτοδυϊκός κώδικας, δηλαδή $C = C^{\perp}$.

3.4.6 Βρείτε έναν τύπο για την απόσταση d^* ενός εκτεταμένου κώδικα C^* συναρτήσει της απόστασης του αρχικού κώδικα C.

3.4.7 Έστω C ένας κώδικας Hamming μήκους 15. Βρείτε το πλήθος των υποδειγμάτων λάθους τα οποία το θεώρημα 1.11.14 μας βεβαιώνει ότι ο εκτεταμένος κώδικας C^* θα ανιχνεύσει και το πλήθος των υποδειγμάτων λάθους τα οποία το θεώρημα 1.12.9 μας βεβαιώνει ότι ο C^* θα διορθώσει. Πόσα υποδείγματα λάθους διορθώνει ο C^*;

3.5 Ο Εκτεταμένος Κώδικας Golay

Σ' αυτήν και στις επόμενες δύο ενότητες θα κατασκευάσουμε και θα αποκωδικοποιήσουμε δύο κώδικες οι οποίοι θα διορθώνουν λάθη με βάρος μικρότερο ή ίσο του 3. Ο εκτεταμένος κώδικας Golay, που θα μελετηθεί σε αυτήν και στην επόμενη ενότητα, πράγματι χρησιμοποιήθηκε στην αποστολή του Voyager, η οποία, στην αρχή του 1980, μας έστειλε πανέμορφες φωτογραφίες του Δία και του Κρόνου.

Έστω B ένας 12×12 πίνακας

$$B = \begin{bmatrix} 110111000101 \\ 101110001011 \\ 011100010111 \\ 111000101101 \\ 110001011011 \\ 100010110111 \\ 000101101111 \\ 001011011101 \\ 010110111001 \\ 101101110001 \\ 011011100011 \\ 111111111110 \end{bmatrix}.$$

Έστω G είναι ο 12×24 πίνακας $G = [I, B]$, όπου I είναι ο 12×12 ταυτοτικός πίνακας. Ο γραμμικός κώδικας C με γεννήτορα πίνακα G ονομάζεται *εκτεταμένος κώδικας Golay* και θα συμβολίζεται με C_{24}.

Για να απομνημονεύσετε τον B, παρατηρείστε ότι ο 11×11 πίνακας B_1 που προέρχεται από τον B όταν σβήσουμε την τελευταία γραμμή και στήλη, έχει κυκλική κατασκευή. Η πρώτη γραμμή του B_1 είναι η 11011100010. Η δεύτερη γραμμή προέρχεται από την πρώτη με μετάθεση κάθε ψηφίου μία θέση αριστερά και φέρνοντας το πρώτο ψηφίο στο τέλος. Η τρίτη γραμμή προέρχεται από την δεύτερη με τον ίδιο τρόπο και συνεχίζουμε έτσι με όλες τις γραμμές. Έτσι ο B μπορεί να απομνημονευτεί ως ο πίνακας

$$B = \begin{bmatrix} B_1 & j^T \\ j & 0 \end{bmatrix},$$

όπου η j είναι η λέξη μήκους 11 και με όλα 1. Με επιθεώρηση βλέπουμε ότι $B^T = B$, δηλαδή ο B είναι ένας συμμετρικός πίνακας.

Τώρα θα παραθέσουμε επτά σημαντικά πράγματα για τον εκτεταμένο κώδικα Golay C_{24} με γεννήτορα πίνακα τον $G = [I, B]$:

(1) Ο C_{24} έχει μήκος $n = 24$, διάσταση $k = 12$ και $2^{12} = 4096$ κωδικολέξεις. Αυτό είναι φανερό μετά από έλεγχο του G.

(2) Ένας parity check πίνακας για το C_{24} είναι ο 24×12 πίνακας

$$\begin{bmatrix} B \\ I \end{bmatrix}$$

και αυτό προκύπτει από τον αλγόριθμο 2.5.7.

(3) Άλλος parity check πίνακας για τον C_{24} είναι ο 24×12 πίνακας

$$H = \begin{bmatrix} I \\ B \end{bmatrix}.$$

Για να το δούμε αυτό, ας παρατηρήσουμε πρώτα ότι κάθε γραμμή του B έχει περιττό βάρος (7 ή 11). Έτσι το εσωτερικό γινόμενο κάθε γραμμής με τον εαυτό της είναι 1. Επίσης με έναν απλό έλεγχο βλέπουμε ότι το εσωτερικό γινόμενο της πρώτης γραμμής του B με οποιαδήποτε άλλη γραμμή είναι 0. Από την κυκλική κατασκευή του B_1, προκύπτει ότι το εσωτερικό γινόμενο δύο οποιονδήποτε διαφορετικών γραμμών του B είναι 0. Οπότε $BB^T = I$. Όμως $B^T = B$, οπότε $B^2 = BB^T$ και

$$GH = [I, B] \begin{bmatrix} I \\ B \end{bmatrix} = I^2 + B^2 = I + BB^T = I + I = 0.$$

Θα χρησιμοποιήσουμε και τους δύο parity check πίνακες για να αποκωδικοποιήσουμε τον C_{24}.

(4) Ακόμα άλλος ένας γεννήτορας πίνακας C_{24} είναι ο 12×24 πίνακας $[B, I]$.

(5) Ο C_{24} είναι αυτοδυϊκός, δηλαδή $C_{24} = C_{24}^{\perp}$.

(6) Η απόσταση του C είναι 8.

(7) Ο C_{24} είναι 3-διόρθωσης λάθους κώδικας.

Οι αποδείξεις των προτάσεων (4) και (5) ζητούνται στις ασκήσεις. Θα αποδείξουμε το (6), το οποίο επιπλέον περιέχει περισσότερες χρήσιμες πληροφορίες για τον κώδικα C_{24}. Η απόδειξη χωρίζετε σε τρία βήματα.

Βήμα I Το βάρος μιας οποιασδήποτε λέξης του C_{24} είναι ένα πολλαπλάσιο του 4. Για να το δούμε αυτό παρατηρούμε κατ'αρχάς ότι οι γραμμές του G έχουν βάρος 8 ή 12. Έστω v είναι η λέξη του C_{24} η οποία είναι το άθροισμα $v = r_i + r_j$ δύο διαφορετικών γραμμών του G. Οι γραμμές του B είναι ορθογώνιες· έτσι οι γραμμές του G είναι ορθογώνιες. Οπότε οι r_i και οι r_j έχουν άρτιο πλήθος, έστω $2x$, των κοινών μονάδων. Οπότε το

$$wt(v) = wt(r_i) + wt(r_j) - 2(2x)$$

είναι ένα πολλαπλάσιο του 4.

Ας υποθέσουμε τώρα ότι η λέξη v του C_{24} είναι το άθροισμα $v = r_i + r_j + r_k$ τριών διαφορετικών γραμμών του G. Έστω $v_1 = r_i + r_j$. Αφού ο C_{24} είναι αυτοδυϊκός, v_1 και r_k έχουν εσωτερικό γινόμενο 0 και έτσι έχουν ένα άρτιο πλήθος, έστω $2y$, των κοινών μονάδων. Έτσι το

$$wt(v) = wt(v_1) + wt(r_k) - 2(2y)$$

είναι πολλαπλάσιο του 4. Συνεχίζοντας έτσι, (ακριβέστερα με επαγωγή), βλέπουμε ότι εάν η v του C_{24} είναι γραμμικός συνδυασμός των γραμμών του G, τότε το $wt(v)$ πρέπει να είναι πολλαπλάσιο του 4.

Βήμα ΙΙ Οι πρώτες 11 γραμμές του G είναι κωδικολέξεις στο C_{24} με βάρος 8, οπότε η απόσταση του C_{24} πρέπει να είναι 4 ή 8.

Βήμα ΙΙΙ Απαλείφουμε λέξεις με βάρος 4 οι οποίες είναι κωδικολέξεις του C_{24}. Έστω v μία μη μηδενική κωδικολέξη στο C_{24} και υποθέτουμε ότι $wt(v) = 4$. Τότε παίρνουμε $v = u_1[I, B]$ και $v = u_2[B, I]$ για κάποιες u_1 και u_2 (διότι και οι δύο, $[I, B]$ και $[B, I]$ γεννούν τον C_{24}) και $wt(u_1) \leq 2$ ή $wt(u_2) \leq 2$ (διότι το μισό του v περιέχει το πολύ 2 άσους). Εντούτοις, κανένα άθροισμα μίας ή δύο γραμμών του B δεν έχει βάρος το πολύ 3, οπότε $wt(v) = wt(u_i) + wt(u_iB) > 4$. Οπότε η v δεν έχει βάρος 4.

Ασκήσεις

3.5.1 Δείξτε ότι η λέξη με μονάδες είναι στο C_{24}. Συμπεράνατε ότι ο C_{24} δεν περιέχει λέξεις με βάρος 20.

3.5.2 Δείξτε το (4) για τον C_{24}.

3.5.3 Δείξτε το (5) για τον C_{24}.

3.5.4 Χρησιμοποιείστε το θεώρημα 2.9.1 για να βεβαιωθείτε ότι ο C_{24} έχει απόσταση 8.

3.6 Αποκωδικοποίηση του εκτεταμένου κώδικα Golay

Θα βρούμε τώρα έναν αλγόριθμο για την ΗΑΜΠ για τον κώδικα C_{24}. Σε όλη την ενότητα, η w θα συμβολίζει την παραληφθείσα λέξη, η v την κοντινότερη κωδικολέξη στην w και με u το υπόδειγμα λάθους $v + w$. Για τον C_{24} θέλουμε να διορθώσουμε όλα τα υποδείγματα με βάρος το πολύ 3, έτσι υποθέτουμε ότι $wt(u) \leq 3$. Ένα κόμμα θα τοποθετηθεί μεταξύ των πρώτων 12 και των τελευταίων 12 ψηφίων κάθε λέξης του K^{24}. Τα υποδείγματα λάθους u θα δηλώνονται με $[u_1, u_2]$, όπου u_1 και u_2 έχουν μήκος το καθένα ίσο με 12. Ο σκοπός μας είναι να ορίσουμε την οδηγό του συμπλόκου, u του συμπλόκου που περιέχει την w χωρίς να πρέπει να αναφερόμαστε στην ΚΠΑ του C_{24}.

Επειδή υποθέτουμε ότι $wt(u) \leq 3$, θα είναι είτε $wt(u_1) \leq 1$ είτε $wt(u_2) \leq 1$. Έστω s_1 είναι το σύνδρομο της $w = v + u$ χρησιμοποιώντας τον parity check πίνακα

$$H = \begin{bmatrix} I \\ B \end{bmatrix}.$$

Τότε $s_1 = wH = [u_1, u_2]H = u_1 + u_2B$. Οπότε εάν $wt(u_2) \leq 1$ τότε η s_1 περιέχει είτε μία λέξη βάρους το πολύ 3 (εάν $wt(u_2) = 0$) είτε μια γραμμή του B με το πολύ 2 από τα ψηφία του αλλαγμένα (εάν $wt(u_2) = 1$). Όμοια, εάν $wt(u_1) \leq 1$ τότε το σύνδρομο

$$s_2 = w \begin{bmatrix} B \\ I \end{bmatrix} = u_1B + u_2$$

περιέχει είτε μία λέξη βάρους το πολύ 3 είτε μία γραμμή του B με το πολύ 2 από τα ψηφία του αλλαγμένα.

Σε κάθε περίπτωση, εάν η u έχει βάρος το πολύ 3 τότε εύκολα το βρίσκουμε, διότι το πολύ 3 γραμμές ενός από τους 2 parity check πίνακες μπορούν να βρεθούν για να προστεθούν στο αντίστοιχο σύνδρομο. Χρησιμοποιώντας αυτήν την παρατήρηση κατασκευάζουμε τον παρακάτω αλγόριθμο αποκωδικοποίησης. Θα χρησιμοποιήσουμε ότι $B^2 = I$ και

$$\begin{aligned} s_1 &= u_1 + u_2B = wH \\ s_2 &= u_1B + u_2 \\ &= (u_1 + u_2B)B = s_1B. \end{aligned}$$

Για να αποφύγουμε τη χρήση και των δύο parity check πινάκων μέσα στον αλγόριθμο θα χρησιμοποιήσουμε μονάχα τον $H = \begin{bmatrix} I \\ B \end{bmatrix}$. Φυσικά μόλις, το u προσδιοριστεί, η w αποκωδικοποιείται στην κωδικολέξη $v = w + u$. e_i είναι η λέξη με μήκος 12 με άσο στην i-οστή θέση και μηδέν αλλού και b_i είναι η i-οστή γραμμή του B.

Αλγόριθμος 3.6.1 ΗΑΜΠ για τον C_{24}).

1. Υπολογίστε το σύνδρομο $s = wH$.
2. Εάν $wt(s) \leq 3$ τότε $u = [s, 0]$.
3. Εάν $wt(s + b_i) \leq 2$ για κάποια γραμμή b_i του B τότε $u = [s + b_i, e_i]$.
4. Υπολογίστε το δεύτερο σύνδρομο sB.
5. Εάν $wt(sB) \leq 3$ τότε $u = [0, sB]$.
6. Εάν $wt(sB + b_i) \leq 2$ για κάποια γραμμή b_i του B τότε $u = [e_i, sB + b_i]$.
7. Εάν η u δεν έχει ακόμα προσδιοριστεί τότε ζητήστε αναμετάδοση.

Ο παραπάνω αλγόριθμος απαιτεί το πολύ 26 υπολογισμούς βαρών στη διαδικασία αποκωδικοποίησης. (Φυσικά, όταν η u προσδιοριστεί, τότε δεν απαιτείται επιπλέον βήματα να γίνουν).

Παράδειγμα 3.6.2 Αποκωδικοποιούμε την $w = 101111101111, 010010010010$. Το σύνδρομο είναι

$$s = wH = 101111101111 + 001111101110$$

$$= 100000000001,$$

το οποίο έχει βάρος 2. Επειδή $wt(s) \leq 3$, βρίσκουμε ότι

$$u = [s, 0] = 100000000001, 000000000000$$

και συμπεραίνουμε ότι η

$$v = w + u = 001111101110, 010010010010$$

είναι η κωδικολέξη που μας αποστάλθηκε .

Επειδή ο $G = [I, B]$ είναι σε κανονική μορφή και κάθε λέξη στο K^{12} μπορεί να κωδικοποιηθεί ως ένα μήνυμα (ο C_{24} έχει διάσταση 12), το μήνυμα που αποστάλθηκε εμφανίζεται στα πρώτα 12 ψηφία της αποκωδικοποιημένης λέξης v. Στο παράδειγμα 3.6.2 το μήνυμα 001111101110 είχε αποσταλλεί.

Παράδειγμα 3.6.3 Αποκωδικοποιούμε την $w = 001001001101, 101000101000$. Το σύνδρομο είναι

$$s = wH = 001001001101 + 111000000100 = 110001001001,$$

το οποίο έχει βάρος 5. Προχωρώντας στο βήμα 3 του αλγορίθμου 3.6.1, υπολογίζουμε τα

$$\begin{aligned} s + b_1 &= 000110001100 \\ s + b_2 &= 011111000010 \\ s + b_3 &= 101101011110 \\ s + b_4 &= 001001100100 \\ s + b_5 &= 000000010010. \end{aligned}$$

Επειδή $wt(s + b_5) \leq 2$, βρίσκουμε ότι

$$u = [s + b_5, e_5] = 000000010010, 000010000000$$

και συμπεραίνουμε ότι η

$$v = w + u = 001001011111, 101010101000$$

είναι η αποσταλθείσα κωδικολέξη.

Παράδειγμα 3.6.4 Αποκωδικοποιούμε την $w = 000111000111, 011011010000$. Το σύνδρομο είναι

$$s = wH = u_1 + u_2B$$
$$= 000111000111 + 101010101101$$
$$= 101101101010,$$

η οποία έχει βάρος 7. Προχωρώντας στο βήμα 3, βρίσκουμε $wt(s + b_i) \geq 3$ για κάθε γραμμή b_i του B. Συνεχίζοντας στο βήμα $4,\cdot$ το δεύτερο σύνδρομο είναι

$$sB = 111001111101,$$

η οποία έχει βάρος 9. Προχωρώντας στο βήμα 5 υπολογίζουμε

$$sB + b_1 = 001110111000$$
$$sB + b_2 = 010111110110$$
$$sB + b_3 = 100101101010$$
$$sB + b_4 = 000001010000.$$

Επειδή $wt(sB + b_4) \leq 2$, βρίσκουμε ότι

$$u = [e_4, sB + b_4] = 000100000000, 000001010000$$

και συμπεραίνουμε ότι

$$v = w + u = 000011000111, 011010000000$$

είναι η κωδικολέξη που μας αποστάλθηκε.

Ασκήσεις

3.6.5 Έστω ο κώδικας C_{24}. Αποκωδικοποιείστε εάν είναι δυνατό κάθε μία από τις παρακάτω παραλειφθήσεις λέξεις w.

(α΄). 111 000 000 000, 011 011 011 011

(β΄). 111 111 000 000, 100 011 100 111

(γ΄). 111 111 000 000, 101 011 100 111

(δ΄). 111 111 000 000, 111 000 111 000

(ε΄). 111 000 000 000, 110 111 001 101

(ϛ΄). 110 111 001 101, 111 000 000 000

(ζ΄). 000 111 000 111, 101 000 101 101

(η΄). 110 000 000 000, 101 100 100 000

(θ΄). 110 101 011 101, 111 000 000 000

3.6.6 Βρείτε το πιο πιθανό υπόδειγμα λάθους για κάθε λέξη με τις παρακάτω συνδρομές

(α'). $s_1 = 010010000000, \quad s_2 = 011111010000$

(β'). $s_1 = 010010100101, \quad s_2 = 001000110000$

(γ'). $s_1 = 111111000101, \quad s_2 = 111100010111$

(δ'). $s_1 = 111111111011, \quad s_2 = 010010001110$

(ε'). $s_1 = 001101110110, \quad s_2 = 111110101101$

(ϛ'). $s_1 = 010111111001, \quad s_2 = 100010111111.$

3.6.7 Δείξτε εάν το s ή το sB έχει βάρος 4 τότε η ΗΑΜΠ απαιτεί την επαναποστολή της λέξεως.

3.7 Ο κώδικας Golay

Ένας άλλος ενδιαφέρον 3- κώδικας διόρθωσης λαθών μπορεί να δημιουργηθεί με *κουτσούρεμα* του C_{24}, δηλαδή με απόσπαση ενός ψηφίου από κάθε λέξη του C_{24}. Το ίδιο ψηφίο πρέπει να αποσπαστεί από κάθε λέξη. Θα αποσπάσουμε το τελευταίο ψηφίο.

Έστω $\hat{B}$ είναι ο 12×11 πίνακας που προέρχεται από τον πίνακα B σβήνοντας την τελευταία στήλη. Έστω G είναι ο 12×23 πίνακας $G = [I_{12}, \hat{B}]$. Ο γραμμικός κώδικας με γεννήτορα πίνακα τον G λέγεται κώδικας Golay και συμβολίζεται με C_{23}. Ο κώδικας Golay έχει μήκος $n = 23$, διάσταση $k = 12$ και περιέχει $2^{12} = 4096$ κωδικολέξεις. Σημειώστε ότι ο εκτεταμένος κώδικας C_{23}^* είναι πράγματι ο C_{24}. Ο C_{23} έχει απόσταση 7. Αυτό φαίνεται εύκολα από το γεγονός ότι $C_{23}^* = C_{24}$ (βλέπε άσκηση 3.4.6), αλλά μπορούμε να δείξουμε χρησιμοποιώντας το θεώρημα 3.2.8 ή με μετατροπή της απόδειξης ότι ο C_{24} έχει απόσταση 8.

Ο κώδικας Golay C_{23} είναι ένας τέλειος κώδικας (παράδειγμα 3.2.4) *και θα διορθώσει όλα τα υποδείγματα λάθους με βάρος* 3 *ή μικρότερο και κανένα άλλο* (θεώρημα 3.2.8). Οπότε κάθε ληφθείσα λέξη w έχει το πολύ απόσταση 3 από ακριβώς μία κωδικολέξη. Οπότε εάν κολλήσουμε το ψηφίο 0 ή 1 στην w σχηματίζοντας την $w0$ ή την $w1$ αντίστοιχα, με τρόπο ώστε η προκύπτουσα λέξη να έχει περιττό βάρος, τότε η προκύπτουσα λέξη έχει απόσταση το πολύ 3 από μία κωδικολέξη c του C_{24} (βλέπε άσκηση 3.7.8). Αποκωδικοποιώντας τη c με τον αλγόριθμο 3.6.1 και αποσπώντας το τελευταίο ψηφίο από τη c, παίρνουμε την πιο κοντινή κωδικολέξη στην w του C_{23}.

Αλγόριθμος 3.7.1 (Αλγόριθμος αποκωδικοποίησης για τον κώδικα Golay).

1. Σχηματίστε την $w0$ ή την $w1$, όποια έχει περιττό βάρος.

2. Αποκωδικοποιείστε wi (i είναι 0 ή 1) χρησιμοποιώντας τον αλγόριθμο 3.6.1 σε μία κωδικολέξη c του C_{24}.

3. Αποσπάστε το τελευταίο ψηφίο από το c.

Στην πράξη η παραληφθείσα λέξη w είναι συνήθως μία κωδικολέξη, όμως η wi που πήραμε στο βήμα 1 δεν είναι ποτέ μία κωδικολέξη (Γιατί;). Εάν η w είναι μία κωδικολέξη τότε το σύνδρομο της wi είναι η τελευταία γραμμή του H (Γιατί;) οπότε αυτό μπορεί εύκολα να διαπιστωθεί πριν ξεκινήσουμε τον αλγόριθμο 3.6.1.

Παράδειγμα 3.7.2 Αποκωδικοποιούμε την $w = 001001001001, 11111110000$. Επειδή η w έχει περιττό βάρος, σχηματίζουμε την $w0 = 001001001001, 111111100000$. Τότε $s_1 = 100010111110$. Επειδή $s_1 = b_6 + e_9 + e_{12}$, η $w0$ κωδικοποιείται ως 001001000000, 111110100000 και άρα η w αποκωδικοποιείται ως 001001000000, 11111010000.

Ασκήσεις

3.7.3 Αποκωδικοποιείστε κάθε μία από τις παρακάτω παραληφθείσες λέξεις που κωδικοποιήθηκαν με χρήση του C_{23}.

(α'). $101011100000,\ \ 10101011011$

(β'). $101010000001,\ \ 11011100010$

(γ'). $100101011000,\ \ 11100010000$

(δ'). $011001001001,\ \ 01101101111$.

3.7.4 Δείξτε ότι ο C_{23} έχει απόσταση $d = 7$.

3.7.5 Βρείτε την αξιοπιστία του C_{23} όταν μεταδίδεται μέσω ενός ΔΣΚ με πιθανότητα p.

3.7.6 Προσδιορίστε ποιος από τους C_{23} ή C_{24} έχει μεγαλύτερη αξιοπιστία. Χρησιμοποιείστε το ίδιο ΔΣΚ και για τους δύο.

3.7.7 Χρησιμοποιείστε το γεγονός ότι κάθε λέξη με βάρος 4 έχει απόσταση 3 από μία ακριβώς κωδικολέξη (γιατί;) για να απαριθμήσετε το πλήθος των κωδικολέξεων βάρους 7 στον κώδικα Golay (Υπόδειξη: Για κάθε κωδικολέξη c, το πλήθος των λέξεων που έχουν βάρος 4 και απόσταση 3 από τη c είναι $\binom{7}{3}$).

3.7.8 Χρησιμοποιείστε την άσκηση 3.7.7, για να δείξετε ότι ο C_{24} περιέχει ακριβώς 759 κωδικολέξεις βάρους 8.

3.7.9 Χρησιμοποιείστε τις ασκήσεις 3.5.1 και 3.7.8 για να διαπιστώσετε τον πίνακα κατανομής βαρών που ακολουθεί:

βάρος	0	4	8	12	16	20	24
αριθμός λέξεων	1	0	759	2576	759	0	1

3.7.10 Έστω w είναι μια παραληφθείσα λέξη που αποκωδικοποιήθηκε με χρήση τον C_{23}. Καλούμε ένα ψηφίο i (0 ή 1) στο w για να σχηματίσουμε μία λέξη wi περιττού βάρους. Δείξτε ότι η wi απέχει απόσταση το πολύ 3 από μία κωδικολέξη του C_{24}. (Υπόδειξη: όλες οι λέξεις στο C_{24} έχουν άρτιο βάρος).

3.8 Κώδικες Reed-Muller

Σ' αυτήν την ενότητα θα θεωρήσουμε μία άλλη σημαντική κλάση κωδίκων που περιέχουν τον εκτεταμένο κώδικα Hamming που συζητήσαμε νωρίτερα. Ο r-οστής τάξης κώδικας Reed-Muller με μήκος 2^m θα συμβολίζεται με $RM(r,m), 0 \leq r \leq m$. Δίνουμε έναν αναδρομικό ορισμό αυτών των κωδίκων

(1) $RM(0,m) = \{00...0, 11...1\}, RM(m,m) = K^{2^m}$

(2) $RM(r,m) = \{(x, x+y) | x \in RM(r,m-1), y \in RM(r-1,m-1)\}, 0 < r < m$.

Οπότε ο $RM(m,m)$ είναι όλες οι λέξεις μήκους 2^m και $RM(0,m)$ είναι η λέξη με όλα άσους (και η μηδενική λέξη).

Παράδειγμα 3.8.1

$$\begin{array}{ll} RM(0,0) = \{0,1\} & \\ RM(0,1) = \{00,11\}, & RM(1,1) = K^2 = \{00,01,10,11\} \\ RM(0,2) = \{0000,1111\}, & RM(2,2) = K^4 \\ RM(1,2) = \{(x, x+y)| & x \in \{00,01,10,11\}, y \in \{00,11\}\} \end{array}$$

Αντί να χρησιμοποιούμε την παραπάνω περιγραφή του κώδικα, θα δώσουμε μ·ια αναδρομική κατασκευή για τον γεννήτορα πίνακα του $RM(r,m)$, ο οποίος θα συμβολίζεται με $G(r,m)$. Για $0 < r < m$, ορίζουμε $G(r,m)$ με

$$G(r,m) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix}$$

Για $r = 0$ ορίζουμε

$$G(0,m) = [11...1]$$

και για $r = m$, ορίζουμε

$$G(m,m) = \begin{bmatrix} G(m-1,m) \\ 0...01 \end{bmatrix}.$$

Παράδειγμα 3.8.2 Οι γεννήτορες πίνακες για τον $RM(0,1)$ και $RM(1,1)$ είναι οι

$$G(0,1) = (1,1) \text{ και } G(1,1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Παράδειγμα 3.8.3 Έστω $m = 2$, τότε το μήκος είναι $4 = 2^2$ και για $r = 1, 2$ έχουμε

$$G(1,2) = \begin{bmatrix} G(1,1) & G(1,1) \\ 0 & G(0,1) \end{bmatrix}, G(2,2) = \begin{bmatrix} G(1,2) \\ 0001 \end{bmatrix}.$$

Χρησιμοποιώντας το παράδειγμα 3.8.2 έχουμε,

$$G(1,2) = \begin{bmatrix} 11 & 11 \\ 01 & 01 \\ 00 & 11 \end{bmatrix}, G(2,2) = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \end{bmatrix}.$$

Παράδειγμα 3.8.4 Για $m = 3$, $m = 2^3 = 8$, έχουμε

$$G(0,3) = (11111111), G(3,3) = \begin{bmatrix} G(2,3) \\ 00000001 \end{bmatrix}$$

$$G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix}, G(2,3) = \begin{bmatrix} G(2,2) & G(2,2) \\ 0 & G(1,2) \end{bmatrix}.$$

Τότε χρησιμοποιώντας το παράδειγμα 3.8.3 έχουμε

$$G(1,3) = \begin{bmatrix} 1111 & 1111 \\ 0101 & 0101 \\ 0011 & 0011 \\ 0000 & 1111 \end{bmatrix}.$$

Ασκήσεις

3.8.5 Βρείτε τον γεννήτορα πίνακα $G(2,3)$.

3.8.6 Βρείτε τον γεννήτορα πίνακα $G(r,4)$ για τους κώδικες $RM(r,4)$ για $r = 0, 1, 2$.

Με τον αναδρομικό ορισμό που είδαμε είναι απλό πράγμα να αποδείξουμε με επαγωγή τις βασικές ιδιότητες ενός κώδικα Reed-Muller.

Θεώρημα 3.8.7 *Ο r-οστής τάξης Reed-Muller κώδικας $RM(r,m)$ που ορίστηκε παραπάνω έχει τις ακόλουθες ιδιότητες:*

(1) μήκος $n = 2^m$

(2) απόσταση $d = 2^{m-r}$

(3) διάσταση $k = \sum_{i=0}^{r} \binom{m}{i}$

(4) ο $RM(r-1,m)$ περιέχεται στο $RM(r,m)$, $r > 0$

(5) ο δυϊκός του κώδικας είναι ο $RM(m-1-r,m)$, $r < m$.

Απόδειξη: Οι αποδείξεις των παραπάνω δηλώσεων χρησιμοποιούν επαγωγή. Το αφήσαμε ως μία άσκηση για να δείξουμε ότι αυτό το θεώρημα ισχύει για όλους τους $RM(r,m)$ κώδικες για $m = 1, 2, 3, 4$. Επίσης σημειώνουμε ότι οι παραπάνω ισχυρισμοί είναι προφανώς αληθινοί για $r = 0$ και $r = m$.

Κατ'αρχάς θέλουμε να δείξουμε ότι $RM(r-1,m) \subseteq RM(r,m)$. Αρχίζουμε με

$$G(1,m) = \begin{pmatrix} G(1,m-1) & G(1,m-1) \\ 0 & G(0,m-1) \end{pmatrix}.$$

Επειδή $\mathbf{1}$ είναι πάνω στη γραμμή του $G(1,m-1)$, τότε το διάνυσμα από άσους $(\mathbf{1},\mathbf{1})$ είναι πάνω στην πρώτη γραμμή του $(G(1,m-1), G(1,m-1))$. Οπότε $RM(0,m) = \{\mathbf{0},\mathbf{1}\}$ είναι υποσύνολο του $RM(1,m)$.

Γενικά επειδή ο $G(r-1,m-1)$ είναι ένας υποπίνακας του $G(r,m-1)$ και ο $G(r-2,m-1)$ είναι ένας υποπίνακας του $G(r-1,m-1)$ έχουμε εύκολα ότι ο

$$G(r-1,m) = \begin{pmatrix} G(r-1,m-1) & G(r-1,m-1) \\ 0 & G(r-2,m) \end{pmatrix}$$

είναι ένας υποπίνακας του $G(r,m)$ και έτσι $RM(r-1,m)$ είναι ένας υποκώδικας του $RM(r,m)$.

Στη συνέχεια βρίσκουμε την απόσταση $d = 2^{m-r}$ για τον $RM(r,m)$, με επαγωγή στο r.

Επειδή $RM(r,m) = \{(x, x+y) | x \in RM(r,m-1), y \in RM(r-1,m-1)\}$ και $RM(r-1,m-1) \subseteq RM(r,m-1)$ τότε $x+y \in RM(r,m-1)$ και άρα εάν $x \neq y$, τότε παίρνουμε από την επαγωγική υπόθεσή μας $wt(x+y) \geq 2^{m-1-r}$. Επίσης $wt(x) \geq 2^{m-1-r}$. Οπότε $wt(x, x+y) = wt(x+y) + wt(x) \geq 2 \cdot 2^{m-1-r} = 2^{m-r}$. Εάν $x = y$, τότε $(x, x+y) = (y, 0)$ όμως $y \in RM(r-1,m-1)$ και έτσι $wt(y,0) = wt(y) \geq 2^{m-1-(r-1)} = 2^{m-r}$.

Από τον ορισμό του $G(r,m)$, έχουμε

$$\begin{aligned} \dim RM(r,m) &= \dim RM(r,m-1) + \dim RM(r-1,m-1) \\ &= \sum_{i=0}^{r} \binom{m-1}{i} + \sum_{i=0}^{r-1} \binom{m-1}{i} \\ &= \sum_{i=1}^{r} \left(\binom{m-1}{i} + \binom{m-1}{i-1} \right) + \binom{m-1}{0}. \end{aligned}$$

Επειδή $\binom{m}{i} = \binom{m-1}{i} + \binom{m-1}{i-1}$ και $\binom{m-1}{0} = 1 = \binom{m}{0}$ έχουμε,

$$\dim RM(r,m) = \sum_{i=0}^{r} \binom{m}{i}.$$

Τελικά έστω

$$RM(r,m) = \{(x, x+y) | x \in RM(r,m-1), y \in RM(r-1,m-1)\}$$

και έστω

$$RM(m-r-1,m) = \{(x', x'+y') | x' \in RM(m-r-1,m-1), y' \in RM(m-r-2,m-1)\}.$$

Με επαγωγή ο δυϊκός του $RM(r,m-1)$ είναι ο $RM(m-r-2,m-1)$ και ο δυϊκός του $RM(r-1,m-1)$ είναι ο $RM(m-r-1,m-1)$ οπότε $x \cdot y' = 0$ και $x' \cdot y = 0$. Επίσης επειδή $RM(r-1,m-1) \subseteq RM(r,m-1)$, $y \cdot y' = 0$. Οπότε

$$\begin{aligned}(x, x+y) \cdot (x', x'+y') &= (x+y) \cdot (x'+y') + x \cdot x' \\ &= 2(x \cdot x') + x \cdot y' + y \cdot x' + y \cdot y' \\ &= 0.\end{aligned}$$

Βλέπουμε ότι κάθε διάνυσμα του $RM(r,m)$ είναι ορθογώνιο με κάθε διάνυσμα του $RM(m-r-1,m)$. Επειδή

$$\begin{aligned}\dim RM(r,m) + \dim RM(m-r-1,m) &= \sum_{i=0}^{r} \binom{m}{i} + \sum_{i=0}^{m-r-1} \binom{m}{i} \\ &= \sum_{i=0} r \binom{m}{m-i} + \sum_{j=0}^{m-r-1} \binom{m}{j} \\ &= \sum_{m}^{j=0} \binom{m}{j} = 2^m\end{aligned}$$

ο $RM(m-r-1,m)$ κώδικας είναι ο δυϊκός του $RM(r,m)$ κώδικα.

Ασκήσεις

3.8.8 Δείξτε ότι το θεώρημα 3.8.7 ισχύει για τους κώδικες $RM(r,m)$, $1 \leq m \leq 4$, που κατασκευάστηκαν στα παραδείγματα 3.8.1, 3.8.3, 3.8.4 και στις ασκήσεις 3.8.5 και 3.8.6.

Θεωρούμε τον κώδικα Reed-Muller πρώτης τάξης $RM(1,m)$. Σημειώστε ότι ο $RM(m-2,m)$ έχει διάσταση $2^m - m - 1$ και έχει απόσταση 4, μήκος 2^m και επομένως είναι ένας εκτεταμένος κώδικας Hamming. Από το θεώρημα 3.8.7 ο $RM(1,m)$ είναι ο δυϊκός αυτού του εκτεταμένου κώδικα Hamming. Θα παρουσιάσουμε έναν αλγόριθμο αποκωδικοποίησης για αυτόν τον κώδικα ο οποίος είναι αρκετά αποτελεσματικός. Για αλγόριθμο αποκωδικοποίησης στη γενική περίπτωση $RM(r,m)$ θα κάνουμε λόγο στο κεφάλαιο 9.

Σημειώστε ότι ο $RM(1,m)$ κώδικας είναι ένας μικρός κώδικας με μεγάλη ελάχιστη απόσταση, οπότε ένας καλός αλγόριθμος αποκωδικοποίησης είναι πράγματι αρκετά στοιχειώδης: για κάθε παραληφθείσα λέξη w, βρείτε την κωδικολέξη του $RM(1,m)$ κοντινότερα στην w. Αυτό μπορεί να γίνει πολύ αποτελεσματικά.

Παράδειγμα 3.8.9 Έστω $m = 3$ και θεωρούμε τον $RM(1, 3)$ κώδικα ο οποίος έχει μήκος $8 = 2^3$ και $16 = 2^{3+1}$ κωδικολέξεις. Η ελάχιστη απόσταση είναι 4. Έστω

$$G(1,3) = \begin{bmatrix} 1111 & 1111 \\ 0101 & 0101 \\ 0011 & 0011 \\ 0000 & 1111 \end{bmatrix}$$

Σημειώστε ότι εάν w παραλήφθηκε και $d(w, c) < 2$ τότε η w αποκωδικοποιείται με c αλλά εάν $d(w, c) > 6$, τότε $d(w, \mathbf{1} + c) < 2$ και η w αποκωδικοποιείται με $\mathbf{1} + c$. (Σημειώστε ότι η $\mathbf{1}$ είναι μία κωδικολέξη). Για παράδειγμα, εάν $w = 10001111$ παραλήφθηκε τότε η $c = 00001111$ είναι η κοντινότερη κωδικολέξη. Εάν η $w = (10101011)$ παραλήφθηκε και βρίσκουμε $c = (01010101)$ με $d(w, c) > 6$, τότε η $c + \mathbf{1} = 10101010$ είναι η κοντινότερη κωδικολέξη. Έτσι πρέπει να ελέγξουμε το πολύ τις μισές λέξεις του $RM(1, m)$.

Πράγματι, υπάρχουν πολύ αποτελεσματικοί μέθοδοι πινάκων για να υπολογίσουμε αυτές τις αποστάσεις αλλά δε θα τις δούμε εδώ.

Ασκήσεις

3.8.10 Έστω $G(1, 3)$ είναι ο γεννήτορας πίνακας για τον $RM(1, 3)$ κώδικα. Προσπαθήστε να αποκωδικοποιήσετε τις παρακάτω παραληφθείσες λέξεις.

(α'). 0101 1110

(β'). 0110 0111

(γ'). 0001 0100

(δ'). 1100 1110.

3.8.11 Έστω $G(1, 4)$ είναι ο γεννήτορας πίνακας για τον $RM(1, 4)$ κώδικα. Αποκωδικοποιείστε τις παρακάτω παραληφθείσες λέξεις

(α'). 1011 0110 0110 1001

(β'). 1111 0000 0101 1111

3.9 Ταχεία αποκωδικοποίηση για τους $RM(1, m)$

Σ' αυτήν την ενότητα παρουσιάζουμε εν συντομία και χωρίς απόδειξη μία πολύ αποτελεσματική αποκωδικοποίηση για τους $RM(1, m)$ κώδικες. Χρησιμοποιεί το γρήγορο μετασχηματισμό Hadamard Transformγια να βρει την κοντινότερη κωδικολέξη. Κατ'αρχάς είναι ανάγκη να εισάγουμε τον πολλαπλασιασμό Kronecher για τους πίνακες.

Ορίζουμε $A \times B = [a_{ij}B]^{\cdot}$ δηλαδή το στοιχείο a_{ij} του A αντικαθίστεται με τον πίνακα $a_{ij}B$.

Παράδειγμα 3.9.1 Έστω $H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ τότε

$$I_2 \times H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$H \times I_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ & & & \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}.$$

Τώρα θα θεωρήσουμε μία ακολουθία από πίνακες που ορίζονται ως εξής:

$$H_m^i = I_{2^{m-i}} \times H \times I_{2^{i-1}}$$

για $i = 1, 2, ..., m$, όπου H είναι όπως στο παράδειγμα 3.9.1.

Παράδειγμα 3.9.2 Έστω $m = 2$. Τότε

$$\begin{aligned} H_2^1 &= I_2 \times H \times I_1 = I_2 \times H \\ H_2^2 &= I_1 \times H \times I_2 = H \times I_2 \end{aligned}$$

(βλέπε και παράδειγμα 3.9.1).

Παράδειγμα 3.9.3 Έστω $m = 3$ τότε

$$H_3^1 = I_4 \times H \times I_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$H_3^2 = I_2 \times H \times I_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

$$H_3^3 = H \times I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & \\ 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Η αναδρομική φύση της κατασκευής των $RM(1,m)$ κωδίκων υποδηλώνει ότι υπάρχει επίσης μια προσέγγιση στην αποκωδικοποίηση. Αυτή είναι η διαισθητική βάση για την κατασκευή του παρακάτω αλγόριθμου αποκωδικοποίησης για τον $RM(1,m)$.

Αλγόριθμος 3.9.4 Υποθέστε ότι η w παραλαμβάνεται και ότι ο $G(1,m)$ είναι ο γεννήτορας πίνακας για τον $RM(1,m)$ κώδικα

(1) αντικαθιστούμε τα 0 και -1 στην w και σχηματίζουμε την $\overline{w}$

(2) υπολογίζουμε τις $w_1 = \overline{w}H_m^1$ και $w_i = w_{i-1}H_m^i$ για $i = 2, 3, ..., m$.

(3) Βρίσκουμε τη θέση j του μεγαλύτερου αριθμού (κατά απόλυτη τιμή) του w_m.

Έστω $v(j) \in K^m$ είναι η δυαδική αναπαράσταση του j (τα ψηφία μικρής τάξης να είναι πρώτα). Τότε εάν η j-οστή θέση του w_m περιέχει θετικό αριθμό, το απεσταλμένο μήνυμα είναι $(1, v(j))$, ενώ εάν περιέχει αρνητικό αριθμό τότε το απεσταλμένο μήνυμα είναι $(0, v(j))$.

Παράδειγμα 3.9.5 Έστω $m = 3$ και $G(1,3)$ είναι ο γεννήτορας πίνακας για τον $RM(1,3)$ (βλέπε παράδειγμα 3.8.9). Εάν η $w = 10101011$ παραλείφθηκε, τότε σχηματίζουμε την $\overline{w} = (1, -1, 1, -1, 1, -1, 1, 1)$. Υπολογίζουμε:

$$\begin{aligned} w_1 &= \overline{w}H_3^1 = (0, 2, 0, 2, 0, 2, 2, 0) \\ w_2 &= w_1 H_3^2 = (0, 4, 0, 0, 2, 2, -2, 2) \\ w_3 &= w_2 H_3^3 = (2, 6, -2, 2, -2, 2, 2, -2) \end{aligned}$$

(δείτε παράδειγμα 3.9.2 για H_3^1, H_3^2, H_3^3).

Το μεγαλύτερο στοιχείο του w_3 είναι 6 και εμφανίζεται στη θέση 1. Επειδή $v(1) = 100$ και $6 > 0$, το απεσταλμένο μήνυμα είναι $m = (1100)$.

Υποθέστε $w = (10001111)$. Τότε $\overline{w} = (1, -1, -1, -1, 1, 1, 1, 1)$ και

$$\begin{aligned} w_1 = \overline{w}H_3^1 &= (0, 2, -2, 0, 2, 0, 2, 0) \\ w_2 = w_1 H_3^2 &= (-2, 2, 2, 2, 4, 0, 0, 0) \\ w_3 = w_2 H_3^2 &= (2, 2, 2, 2, -6, 2, 2, 2) \end{aligned}$$

το μεγαλύτερο στοιχείο του w είναι -6 και εμφανίζεται στη θέση 4. Επειδή $v(4) = 001$ και $-6 < 0$ το απεσταλμένο μήνυμα είναι $m = (0001)$.

Ασκήσεις

3.9.6 Αποκωδικοποιείστε τις παραληφθείσες λέξεις στην άσκηση 3.8.10 χρησιμοποιώντας τον αλγόριθμο 3.9.4 (και το παράδειγμα 3.9.2).

3.9.7 Υπολογίστε τα H_4^i για $i = 1, 2, 3, 4$.

3.9.8 Αποκωδικοποιείστε τις παραληφθείσες λέξεις στην άσκηση 3.8.11 χρησιμοποιώντας τον αλγόριθμο 3.9.4 (και την άσκηση 3.9.6).

Κεφάλαιο 4

Κυκλικοί Γραμμικοί Κώδικες

4.1 Πολυώνυμα και Λέξεις

Θα μας είναι χρήσιμο να αναπαραστήσουμε τους κυκλικούς κώδικες με την βοήθεια των πολυωνύμων. Γι' αυτό το λόγο θα αναφέρουμε, εν συντομία, κάποιες αναγκαίες γνώσεις για πολυώνυμα (μίας μεταβλητής).

Ένα *πολυώνυμο βαθμού n υπέρ του K* είναι ένα πολυώνυμο της μορφής $a_0 + a_l x + ... + a_n x^n$ όπου οι συντελεστές $a_0, ..., a_n$ είναι στοιχεία του K. Το σύνολο όλων των πολυωνύμων υπέρ του K συμβολίζεται με $K[x]$. Τα στοιχεία του $K[x]$ θα συμβολίζονται με $f(x)$, $g(x)$, $p(x)$ κ.ο.κ.

Τα πολυώνυμα υπέρ του K προστίθονται και πολλαπλασιάζονται ως συνήθως λαμβάνοντας υπόψη ότι, επειδή $1+1=0$, έχουμε ότι $x^k + x^k = 0$. Αυτό σημαίνει ότι ο βαθμός του $f(x)+g(x)$ δεν είναι απαραίτητα ο $\max\{\deg f(x), \deg g(x)\}$.

Παράδειγμα 4.1.1 Έστω $f(x) = 1 + x + x^3 + x^4$, $g(x) = x + x^2 + x^3$ και $h(x) = 1 + x^2 + x^4$. Τότε:

(α'). $f(x) + g(x) = 1 + x^2 + x^4$.

(β'). $f(x) + h(x) = x + x^2 + x^3$.

(γ').

$$\begin{aligned} f(x)g(x) &= (x + x^2 + x^3) + x(x + x^2 + x^3) + x^3(x + x^2 + x^3) + \\ &\quad x^4(x + x^2 + x^3) \\ &= x + x^7. \end{aligned}$$

Ασκήσεις

4.1.2 Βρείτε το άθροισμα και το γινόμενο σε κάθε ένα από τα παρακάτω ζεύγη πολυωνύμων υπέρ του K:

(α'). $f(x) = x^5 + x^6 + x^7$, $h(x) = 1 + x^2 + ^3 + x^4$,

(β'). $f(x) = 1 + x^2 + x^3 + x^8 + x^{13}$, $h(x) = 1 + x^3 + x^9$,

(γ'). $f(x) = 1 + x$, $h(x) = 1 + x + x^2 + x^3 + x^4$.

4.1.3 Έστω $f(x) = 1 + x$. Βρείτε:

(α'). $(f(x))^2$

(β'). $(f(x))^3$

(γ'). $(f(x))^4$.

4.1.4 Επαναλάβετε την άσκηση 4.1.3 για $f(x) = 1 + x + x^2$.

4.1.5 Βρείτε όλα τα πολυώνυμα υπέρ του K βαθμού n, για $n = 0$, $n = 2$, $n = 3$ και $n = 4$.

4.1.6 Βρείτε τον αριθμό των πολυωνύμων υπέρ του K βαθμού το πολύ 10.

4.1.7 Ίσως θα έχετε ήδη παρατηρήσει στις ασκήσεις 4.1.3 (α') και 4.1.4 (α'), για κάθε πολυώνυμο $f(x)$ και $g(x)$ του $K[x]$,

$$(f(x) + g(x))^2 = (f(x))^2 + (g(x))^2$$

επειδή $x^k + x^k = 0$. Υπάρχει κάποιος κανόνας στο $K[x]$ για

(α'). $(f(x) + g(x))^4$,

(β'). $(f(x) + g(x))^3$,

(γ'). $(f(x) + g(x))^n$, για κάθε θετικό ακέραιο, n;

Ο συνηθισμένος αλγόριθμος της διαίρεσης δουλεύει για πολυώνυμα υπέρ του K, όπως ακριβώς για τα πολυώνυμα υπέρ των ρητών αριθμών.

Αλγόριθμος 4.1.8 (Ο Αλγόριθμος της Διαίρεσης.) Έστω $f(x)$ και $h(x)$ είναι πολυώνυμα του $K[x]$ με $h(x) \neq 0$. Τότε υπάρχουν μοναδικά πολυώνυμα $q(x)$ και $r(x)$ του $K[x]$ έτσι ώστε

$$f(x) = q(x)h(x) + r(x),$$

με $r(x) = 0$ ή βαθμός $(r(x)) <$ βαθμός $(h(x))$.

Το πολυώνυμο $q(x)$ ονομάζεται *πηλίκο*, και το $r(x)$ ονομάζεται *υπόλοιπο*. Ο αλγόριθμος της εύρεσης του πηλίκου και του υπόλοιπου όταν το $h(x)$ διαιρείται από το $f(x)$ είναι ο γνωστός μας μακρύς αλγόριθμος της διαίρεσης, αλλά με αριθμητική του K μεταξύ των συντελεστών.

Παράδειγμα 4.1.9 Έστω $f(x) = x+x^2+x^6+x^7+x^8$ και $h(x) = 1+x+x^2+x^4$. Τότε

$$\begin{array}{l|l} x^8+x^7+x^6+x^2+x & x^4+x^2+x+1 \\ \cline{2-2} x^8+x^6+x^5+x^4 & x^4+x^3 \\ \cline{1-1} x^7+x^5+x^4+x^2+x & \\ x^7+x^5+x^4+x^3 & \\ \cline{1-1} x^3+x^2+x & \end{array}$$

Οπότε το πηλίκο είναι το $q(x) = x^3+x^4$ και το υπόλοιπο είναι το $r(x) = x+x^2+x^3$. Μπορούμε να γράψουμε $f(x) = h(x)(x^3+x^4)+(x+x^2+x^3)$. Σημειώστε ότι ο βαθμός $(r(x)) <$ βαθμός $(h(x)) = 4$.

Ασκήσεις

4.1.10 Βρείτε το πηλίκο και το υπόλοιπο όταν το $f(x)$ διαιρείται με το $h(x)$ για κάθε ένα από τα ζεύγη υπέρ του K στην άσκηση 4.1.2.

4.1.11 Βρείτε το πηλίκο και το υπόλοιπο σε κάθε περίπτωση όταν το $f(x)$ διαιρείται με το $h(x)$.

(α'). $f(x) = x^2+x^3+x^4+x^8$, $h(x) = 1+x^5$.

(β'). $f(x) = 1+x^{10}$, $h(x) = 1+x^5$.

(γ'). $f(x) = 1+x^7$, $h(x) = 1+x+x^3$.

(δ'). $f(x) = 1+x^{15}$, $h(x) = 1+x^4+x^5+x^7+x^8$.

Το πολυώνυμο $f(x) = a_0+a_lx+a_2x^2+...+a_{n-1}x^{n-1}$ με βαθμό το πολύ $n-1$ υπέρ του K μπορεί να θεωρηθεί ως η λέξη $v = a_0a_1a_2...a_{n-1}$ μήκους n του K^n. Για παράδειγμα εάν $n = 7$,

πολυώνυμο	λέξη
$1+x+x^2+x^4$	1110100
$1+x^4+x^5+x^6$	1000111
$1+x+x^3$	1101000

Έτσι *ένας κώδικας C μήκους n μπορεί να αναπαρασταθεί ως το σύνολο των πολυωνύμων υπέρ του K βαθμού το πολύ $n-1$.*

Σημειώστε ότι είναι βολικότερο για λόγους αναπαραστάσεως λέξεων με πολυώνυμα να απαριθμούμε τα ψηφία μιας λέξης μήκους n από το 0 έως το $n-1$, παρά από το 1 έως το n. Για παράδειγμα η λέξη $a_0a_1a_2a_3$ μήκους 4 αναπαρίσταται με το πολυώνυμο $a_0+a_1x+a_2x^2+a_3x^3$ βαθμού 3.

Παράδειγμα 4.1.12 Ο κώδικας C στην αριστερή στήλη του πίνακα αναπαρίσταται από τα πολυώνυμα στη δεξιά στήλη.

κωδικολέξη c	πολυώνυμο $c(x)$
0000	0
1010	$1+x^2$
0101	$x+x^3$
1111	$1+x+x^2+x^3$

Ασκήσεις

4.1.13 Αναπαραστήστε με πολυώνυμα κάθε κωδικολέξη του C στους παρακάτω κώδικες.

(α'). $C=\{000,001,010,011\}$

(β'). $C=\{000,001,010,011\}$

(γ'). $C=\{0000,0001,1110\}$

(δ'). $C=\{0000,1001,0110,1111\}$

(ε'). $C=\{00000,11111\}$

(ϛ'). $C=\{00000,11100,00111,11011\}$.

4.1.14 Γράψτε όλα τα στοιχεία του κώδικα Hamming μήκους 7 που γεννιέται από τον πίνακα G και στη συνέχεια να τον αναπαραστήσετε με πολυώνυμα.

$$G=\begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}.$$

Στην άσκηση 4.1.11 (α'), ο αναγνώστης υπολόγισε το υπόλοιπο $r(x)$ όταν το $f(x)=x^2+x^3+x^4+x^8$ διαιρείται με το $h(x)=1+x^5$. Το αποτέλεσμα ήταν $r(x)=x^2+x^4$. Από τον Αλγόριθμο της Διαίρεσης το $r(x)$ είναι μοναδικό. Επίσης το $r(x)$ έχει βαθμό μικρότερο από το βαθμό του διαιρέτη $h(x)$.

Λέμε ότι το $f(x)$ *modulo* $h(x)$ είναι το $r(x)$ εάν το $r(x)$ είναι υπόλοιπο της διαίρεσης του $f(x)$ με το $h(x)$· θα γράφουμε $r(x)=f(x) \bmod h(x)$. Επίσης, λέμε ότι δύο συναρτήσεις $f(x)$ και $p(x)$ είναι *ισοδύναμες modulo* $h(x)$ εάν και μόνο εάν έχουν το ίδιο υπόλοιπο όταν διαιρεθούν με $h(x)$. Δηλαδή εάν

$$f(x) \bmod h(x) = r(x) = p(x) \bmod h(x).$$

Το συμβολίζουμε αυτό με

$$f(x) \equiv p(x) \pmod{h(x)}.$$

Παράδειγμα 4.1.15 Έστω $h(x) = 1 + x^5$ και $f(x) = 1 + x^4 + x^9 + x^{11}$. Τότε διαιρώντας το $f(x)$ με το $h(x)$ δίνει υπόλοιπο $r(x) = 1 + x$. Λέμε ότι $r(x) = f(x) \bmod h(x)$.

Ομοίως, εάν $p(x) = 1 + x^6$, τότε $1 + x = 1 + x^6 \bmod (1 + x^5)$ και έτσι λέμε $p(x) \equiv f(x) \bmod h(x)$.

Παράδειγμα 4.1.16 Έστω $h(x) = 1 + x^2 + x^5$. Υπολογίζοντας $f(x) \bmod h(x)$, με $f(x) = 1 + x^2 + x^6 + x^9 + x^{11}$ παίρνουμε υπόλοιπο $r(x) = x + x^4$ και έτσι $x + x^4 = f(x) \bmod h(x)$. Σημειώστε ότι εάν $p(x) = x^2 + x^8$ τότε $p(x) \bmod h(x) = 1 + x^3$ και άρα τα $p(x)$ και $f(x)$ δεν είναι ισοδύναμα $\bmod h(x)$.

Η πρόσθεση και ο πολλαπλασιασμός των πολυωνύμων «σέβεται» την ισοδυναμία των πολυωνύμων όπως την ορίσαμε παραπάνω . Δηλαδή ισχύει:

Λήμμα 4.1.17 *Εάν* $f(x) \equiv g(x) \bmod h(x)$ *τότε,*

$$\begin{aligned} & f(x) + p(x) \equiv g(x) + p(x) \pmod{h(x)} \\ \textit{και} \quad & f(x)p(x) \equiv g(x)p(x) \pmod{h(x)} \end{aligned}$$

Απόδειξη: Υποθέτουμε $r(x) = f(x) \bmod h(x)$ και $r(x) = g(x) \bmod h(x)$ και $s(x) = p(x) \bmod h(x)$ τότε έχουμε

$$\begin{aligned} f(x) + p(x) &= q_1(x)h(x) + r(x) + q_2(x)h(x) + s(x) \\ &= (q_1(x) + q_2(x))h(x) + r(x) + s(x). \end{aligned}$$

Οπότε $r(x) + s(x) = (f(x) + p(x)) \bmod h(x)$ διότι ο βαθμός του $r(x) + s(x) <$ βαθμός $h(x)$ (Γιατί;). Δουλεύοντας όμοια βλέπουμε ότι $r(x) + s(x) = (g(x) + p(x)) \bmod h(x)$. Αφήνουμε τη συνέχεια της απόδειξης στην άσκηση 4.1.22.

Παράδειγμα 4.1.18 Έστω $h(x) = 1 + x^5$, $f(x) = 1 + x + x^7$, $g(x) = 1 + x + x^2$ και $p(x) = 1 + x^6$· οπότε $f(x) \equiv g(x) \pmod{h(x)}$. Τότε

$$f(x) + p(x) = x + x^6 + x^7$$

και

$$g(x) + p(x) = x + x^2 + x^6$$

αλλά

$$(x + x^6 + x^7) \bmod h(x) = x^2 = (x + x^2 + x^6) \bmod h(x).$$

Όμοια

$$(1 + x + x^7)(1 + x^6) \bmod h(x) = 1 + x^3 = (1 + x + x^2)(1 + x^6) \bmod h(x).$$

Σημειώστε ότι $1 + x = (1 + x^6) \bmod h(x)$. Έτσι έχουμε

$$\begin{aligned} (1 + x + x^7)(1 + x^6) &\equiv (1 + x + x^2)(1 + x^6) \\ &\equiv (1 + x + x^2)(1 + x) \equiv 1 + x^3 \pmod{h(x)}. \end{aligned}$$

Ασκήσεις

4.1.19 Έστω $h(x) = 1 + x^3 + x^5$. Υπολογίστε το $f(x) \bmod h(x)$ και την αντίστοιχή του λέξη:

(α'). $f(x) = 1 + x + x^6$

(β'). $f(x) = x + x^4 + x^7 + x^8$

(γ'). $f(x) = 1 + x^{10}$

4.1.20 Έστω $h(x) = 1 + x^7$. Υπολογίστε τα $f(x) \bmod h(x)$ και $p(x) \bmod h(x)$ και αποφασίστε εάν $f(x) \equiv p(x) \bmod h(x)$:

(α'). $f(x) = 1 + x^3 + x^8$, $p(x) = x + x^3 + x^7$

(β'). $f(x) = x + x^5 + x^9$, $p(x) = x + x^5 + x^6 + x^{13}$

(γ'). $f(x) = 1 + x$, $p(x) = x + x^7$

4.1.21 Έστω $h(x) = 1+x^7$ υπολογίστε τα $(f(x)+g(x)) \bmod h(x)$ και $(f(x)g(x)) \bmod h(x)$, όπου

(α'). $f(x) = 1 + x^6 + x^8$, $g(x) = 1 + x$

(β'). $f(x) = 1 + x^5 + x^9$, $g(x) = x + x^2 + x^7$

(γ'). $f(x) = 1 + x^4 + x^5$, $g(x) = 1 + x + x^2$

4.1.22 Αποδείξτε ότι εάν $f(x) \equiv g(x) \pmod{h(x)}$ τότε $f(x)p(x) \equiv g(x)p(x) \pmod{h(x)}$.

4.2 Εισαγωγή στους Κυκλικούς Κώδικες

Τώρα ξεκινάμε τη μελέτη μιας κατηγορίας κωδίκων, που ονομάζονται κυκλικοί κώδικες. Τελικά θα είμαστε σε θέση να χρησιμοποιήσουμε τις γνώσεις μας στους κυκλικούς κώδικες για να κατασκευάσουμε τον γεννήτορα πίνακα για 2- κώδικες BCH διόρθωσης λαθών, όπως και σε κάποιους άλλους κώδικες. Πράγματι θα δούμε επίσης ότι οι κώδικες Hamming και Golay είναι κυκλικοί κώδικες ή είναι ισοδύναμοι με κυκλικούς κώδικες.

Έστω v μία λέξη μήκους n. Η *κυκλική μετάθεση* $\pi(v)$, του v είναι η λέξη μήκους n που παίρνουμε από τη v με μετάθεση του τελευταίου ψηφίου της v στην αρχή και όλα τα υπόλοιπα ψηφία να μετατίθονται μία θέση προς τα δεξιά. Για παράδειγμα:

v	10110	111000	0000	1011
$\pi(v)$	01011	011100	0000	1101

Ένας κώδικας C ονομάζεται κυκλικός κώδικας εάν η κυκλική μετάθεση κάθε κωδικολέξης είναι επίσης μία κωδικολέξη.

Παράδειγμα 4.2.1 Ο κώδικας $C = \{000, 110, 101, 011\}$ είναι ένας γραμμικός κυκλικός κώδικας. Πρώτον ο C είναι γραμμικός. Στη συνέχεια υπολογίζουμε τις $\pi(v)$ για όλες τις v του C.

$$\pi(000) = 000, \pi(110) = 011, \pi(101) = 110, \pi(011) = 101.$$

Επειδή η $\pi(v)$ ανήκει επίσης στον C, για κάθε v του C, ο C είναι γραμμικός.

Παράδειγμα 4.2.2 Ο κώδικας $C = \{000, 100, 011, 111\}$ δεν είναι γραμμικός. Η κυκλική μετάθεση της $v = 100$ είναι η $\pi(100) = 010$ που δεν ανήκει στον C.

Σημειώστε ότι η κυκλική μετάθεση π είναι ένας γραμμικός μετασχηματισμός· δηλαδή,

Λήμμα 4.2.3 $\pi(v + w) = \pi(v) + \pi(w)$ *και* $\pi(av) = a\pi(v)$, $a \in K = \{0, 1\}$. *Έτσι για να δείξουμε ότι ένας γραμμικός κώδικας C είναι κυκλικός είναι αρκετό να δείξουμε ότι $\pi(v) \in C$ για κάθε λέξη v που ανήκει σε μία βάση του C.*

Απόδειξη: Έστω $v = (v_0 v_1 ... v_{n-1})$, $w = (w_0 w_1 ... w_{n-1})$ τότε $v + w = (v_0 + w_0, v_1 + w_1, ..., v_{n-1} + w_{n-1})$ και $\pi(v + w) = (v_{n-1} + w_{n-1}, v_0 + w_0, ..., v_{n-2} + w_{n-2}) = \pi(v) + \pi(w)$.

Παράδειγμα 4.2.4 Στο παράδειγμα 4.2.1, $\{110, 101\}$ είναι η βάση του C και επειδή $\pi(110) = 011$ και $\pi(101) = 110$ ανήκουν στο C, ο C είναι ένας γραμμικός κυκλικός κώδικας.

Εάν επιθυμούμε να κατασκευάσουμε ένα γραμμικό κώδικα, τότε επιλέγουμε μία λέξη v, σχηματίζουμε ένα σύνολο S που αποτελείται από την v και όλες τις κυκλικές μεταθέσεις του, $S = \{v, \pi(v), \pi^2(v), ..., \pi^{n-1}(v)\}$ και ορίζουμε τον C να είναι ο γραμμικός χώρος που παράγεται από το S. Δηλαδή $C = \langle S \rangle$. (Χρησιμοποιούμε το συμβολισμό $\pi^2(v) = \pi(\pi(v))$, $\pi^3(v) = \pi(\pi(\pi(v)))$, κ.ο.κ.) Επειδή το S περιέχει μία βάση του C, ο C πρέπει να είναι κυκλικός από το λήμμα 4.2.3.

Παράδειγμα 4.2.5 Έστω $n = 3$ και $v = 100$. Τότε $S = \{v, \pi(v), \pi^2(v)\} = \{100, 010, 001\}$ και $\langle S \rangle = K^3$. Σημειώστε ότι εάν $w = a_0 v + a_1 \pi(v) + a_2 \pi^2(v)$ τότε $\pi(w) = a_0 \pi(v) + a_1 \pi^2(v) + a_2 \pi^3(v) = a_2 v + a_0 \pi(v) + a_1 \pi^2(v)$.

Παράδειγμα 4.2.6 Έστω $n = 4$ και $v = 0101$. Τότε $\pi(v) = 1010$ και $\pi^2(v) = 0101 = v$. Οπότε $S = \{0101, 1010\}$ και ο $C = \langle S \rangle$ είναι ο κυκλικός κώδικας, $C = \{0000, 0101, 1010, 1111\}$.

Εάν μια λέξη v και οι κυκλικές της μεταθέσεις σχηματίζουν ένα σύνολο $S = \{v, \pi(v), ..., \pi^{n-1}(v)\}$ το οποίο παράγει τον κώδικα C (δηλ. $C = \langle S \rangle$), τότε λέμε ότι η v είναι ένας *γεννήτορας* του γραμμικού κυκλικού κώδικα C. Επειδή κάθε γραμμικός κυκλικός κώδικας που περιέχει την v πρέπει να περιέχει και το S, λέμε ότι ο C είναι ο μικρότερος γραμμικός κυκλικός κώδικας που περιέχει τη v. Αξίζει να σημειωθεί ότι ένας γραμμικός κυκλικός κώδικας μπορεί να έχει πολλούς γεννήτορες.

Ασκήσεις

4.2.7 Βρείτε τη βάση για τον μικρότερο γραμμικό κυκλικό κώδικα μήκους n, που περιέχει τη v:

(α'). $v = 1101000$, $n = 7$

(β'). $v = 010101$, $n = 6$

(γ'). $v = 11011000$, $n = 8$

4.2.8 Βρείτε όλες τις λέξεις v μήκους n, έτσι ώστε $\pi(v) = v$.

4.2.9 Βρείτε όλες τις λέξεις v μήκους 6 έτσι ώστε

(α'). $\pi^2(v) = v$

(β'). $\pi^3(v) = v$.

Οι κυκλικοί κώδικες έχουν μία slick αναπαράσταση ως πολυώνυμα. Αυτό βασίζεται στην απλή παρατήρηση ότι εάν η λέξη v αντιστοιχεί στο πολυώνυμο $v(x)$ τότε η κυκλική μετάθεση του v, $\pi(v)$ αντιστοιχεί στο πολυώνυμο $xv(x) \bmod 1+x^n$. Σημειώστε ότι γενικά $1 \equiv x^n \pmod{1+x^n}$.

Παράδειγμα 4.2.10 Έστω $v = 100$ τότε $v(x) = 1$ και η $\pi(v) = 010$ αντιστοιχεί στο $xv(x) = x$. Όμοια εάν $v = 1101$ τότε $v(x) = 1 + x + x^3$ και $\pi(v) = 1110$ αντιστοιχεί στο $xv(x) \bmod 1 + x^4 = 1 + x + x^3$.

Στους κυκλικούς κώδικες αναφερόμαστε στα στοιχεία του κώδικα ως κωδικολέξεις και ως πολυώνυμα. Τώρα μπορούμε να επαναλάβουμε την προηγούμενη συζήτηση στους κυκλικούς κώδικες χρησιμοποιώντας πολυώνυμα. Δοθέντος μίας λέξης v μήκους n, έστω $v(x)$ το πολυώνυμο που αντιστοιχεί σ' αυτήν. Τότε οι κυκλικές μεταθέσεις της v αντιστοιχούν στα πολυώνυμα $x^i v(x) \bmod 1 + x^n$ για $i = 0, 1, ..., n-1$.

Παράδειγμα 4.2.11 Έστω $v = 1101000$ και $n = 7$. Τότε $v(x) = 1 + x + x^3$ και

λέξη	πολυώνυμο $\pmod{1+x^7}$	
0110100	$xv(x) = x + x^2 + x^4$	
0011010	$x^2v(x) = x^2 + x^3 + x^5$	
0001101	$x^3v(x) = x^3 + x^4 + x^6$	
1000110	$x^4v(x) = x^4 + x^5 + x^7$	$\equiv 1 + x^4 + x^5 \pmod{1+x^7}$
0100011	$x^5v(x) = x^5 + x^6 + x^8$	$\equiv x + x^5 + x^6 \pmod{1+x^7}$
1010001	$x^6v(x) = x^6 + x^7 + x^9$	$\equiv 1 + x^2 + x^6 \pmod{1+x^7}$

Είναι φανερό ότι εάν $c(x) \in \langle\{v(x), xv(x), ..., x^{n-l}v(x)\}\rangle, \pmod{1+x^n}$ τότε αυτό σημαίνει ότι

$$\begin{aligned} c(x) &= (a_0 v(x) + a_1 xv(x) + ... + a_{n-1}x^{n-1}v(x)) \bmod 1 + x^n \\ &= (a_0 + a_1 x + a_2 x^2 + ... + a_{n-1}x^{n-1})v(x) \bmod 1 + x^n \\ &= a(x)v(x) \bmod 1 + x^n \end{aligned}$$

Οπότε παίρνουμε το παρακάτω αποτέλεσμα.

Λήμμα 4.2.12 *Έστω C είναι ένας κυκλικός κώδικας και έστω $v \in C$. Τότε για κάθε πολυώνυμο $a(x)$, $c(x) = a(x)v(x) \bmod (1+x^n)$ είναι μία κωδικολέξη στο C.*

Μεταξύ όλων των μη μηδενικών κωδικολέξεων σε ένα γραμμικό κυκλικό κώδικα C, υπάρχει μία μοναδική λέξη $g \in C$, έτσι ώστε το $g(x)$ να έχει ελάχιστο βαθμό, όπως δείχνει η απόδειξη που ακολουθεί. Στα σίγουρα υπάρχει τουλάχιστον μία λέξη ή ένα πολυώνυμο με τον μικρότερο βαθμό στο C. Εάν δύο μη μηδενικές λέξεις g και g' αντιστοιχούν στα πολυώνυμα $g(x)$ και $g'(x)$ ελάχιστου βαθμού k τότε το $g(x)+g'(x) = c(x) \in C$ διότι ο C είναι γραμμικός και ο βαθμός του $(c(x)) < k$ (επειδή $x^k + x^k = 0$). Επειδή η g είναι μία μη μηδενική λέξη ελάχιστου βαθμού, ο βαθμός του $(c(x)) < k$ σημαίνει ότι $c(x) = 0$, οπότε $g(x) = g'(x)$ και έτσι το $g(x)$ είναι μοναδικό.

Ορίζουμε το γεννήτορα πολυώνυμο(ή παράγων πολυώνυμο generator polynomial) ενός γραμμικού κώδικα C να είναι το μοναδικό μη μηδενικό πολυώνυμο ελάχιστου βαθμού στο C. Από την προηγούμενη συζήτηση ξέρουμε ότι είναι μοναδικό, αλλά είναι ένας γεννήτορας;

Για να δούμε ότι πράγματι είναι πρέπει να δείξουμε ότι για κάθε $c(x) \in C$, υπάρχει ένα $a(x)$ έτσι ώστε, $c(x) = a(x)g(x) \bmod 1+x^n$· πράγματι θα δείξουμε ότι $c(x) = a(x)g(x)$. Επειδή ο βαθμός του $(c(x)) \geq$ του βαθμού του $(g(x))$ έχουμε από τον Αλγόριθμο της Διαίρεσης,

$$c(x) = q(x)g(x) + r(x)$$

ή

$$r(x) = q(x)g(x) + c(x).$$

Όμως και το $c(x)$ και το $q(x)g(x)$ είναι κωδικολέξεις του C από το λήμμα 4.2.12 και άρα και το $r(x)$. Αλλά από τον Αλγόριθμο της Διαίρεσης έχουμε είτε $r(x) = 0$ ή ο βαθμός του $(r(x)) <$ του βαθμού του $(g(x))$. Επειδή το τελευταίο είναι αδύνατο εκτός και εάν $r = 0$, συμπεραίνουμε ότι $r(x) = 0$ και άρα το $g(x)$ είναι ο διαιρέτης κάθε κωδικολέξης $c(x)$ του C.

Θεώρημα 4.2.13 *Έστω C είναι ένας κυκλικός κώδικας μήκους n και έστω $g(x)$ είναι το πολυώνυμο -γεννήτορας. Εάν $n-k =$ ο βαθμός του $(g(x))$ τότε*

(1) ο C έχει διάσταση k,

(2) οι κωδικολέξεις που αντιστοιχούν στα $g(x), xg(x), ..., x^{k-1}g(x)$ αποτελούν βάση του C και

(3) το $c(x) \in C$ εάν και μόνο εάν $c(x) = a(x)g(x)$ για κάποιο πολυώνυμο $a(x)$ με βαθμό $(a(x)) < k$ (δηλαδή, $g(x)$ είναι ο διαιρέτης για κάθε κωδικολέξη $c(x)$).

Απόδειξη: Η συζήτηση που κάναμε πριν από το θεώρημα 4.2.13 αποδεικνύει το (3). Εάν το $g(x)$ έχει βαθμό $n-k$ τότε τα $g(x), xg(x), ..., x^{k-1}g(x)$ πρέπει να είναι γραμμικώς ανεξάρτητα (Γιατί;). Επειδή το $g(x)$ διαιρεί κάθε κωδικολέξη,

υπάρχει ένα μοναδικό πολυώνυμο $a(x) = a_0 + a_1x + ... + a_{k-1}x^{k-1}$ έτσι ώστε $c(x) = a(x)g(x) = a_0g(x) + a_1xg(x) + ... + a_{k-1}x^{k-1}g(x)$. Οπότε το $c(x)$ ανήκει στο $\langle\{g(x), xg(x), ..., x^{k-1}g(x)\}\rangle$ και έτσι το $\{g(x), xg(x), ..., x^{k-1}g(x)\}$ είναι βάση του C.

Παράδειγμα 4.2.14 Έστω $n = 7$, $g(x) = 1 + x + x^3$ ένας γεννήτορας για τον κυκλικό κώδικα C. Μία βάση του C είναι

$$\begin{array}{ll} g(x) & = 1 + x + x^3 \leftrightarrow 1101000 \\ xg(x) & = x + x^2 + x^4 \leftrightarrow 0110100 \\ x^2g(x) & = x^2 + x^3 + x^5 \leftrightarrow 0011010 \\ x^3g(x) & = x^3 + x^4 + x^6 \leftrightarrow 0001101 \end{array}$$

Σημειώστε ότι το $x^4g(x) \bmod 1 + x^7 = 1 + x^4 + x^5$ είναι μία κωδικολέξη επειδή $1 + x^4 + x^5 = (1 + x + x^2)(1 + x + x^3) = (1 + x + x^2)g(x)$.

Παράδειγμα 4.2.15 Έστω C να είναι ο κυκλικός κώδικας $C = \{10000, 1010, 0101, 1111\}$· τα αντίστοιχα πολυώνυμα είναι $\{0, 1 + x^2, x + x^3, 1 + x + x^2 + x^3\}$. Παρατηρήστε ότι το, $1 + x^2 \leftrightarrow 1010$ είναι το πολυώνυμο -γεννήτορας του C, αφού ο C περιέχει μόνο ένα πολυώνυμο βαθμού 2 και κανένα βαθμού 1. Επίσης, κάθε λέξη (πολυώνυμο) στο C είναι ένα πολλαπλάσιο του πολυωνύμου γεννήτορα:

$$\begin{array}{llll} 0 & = 0(1 + x^2) & x + x^3 & = x(1 + x^2) \\ 1 + x^2 & = 1(1 + x^2) & 1 + x + x^2 + x^3 & = (1 + x)(1 + x^2). \end{array}$$

Παράδειγμα 4.2.16 Ο μικρότερος γραμμικός κυκλικός κώδικας C μήκους 6 που περιέχει το $g(x) = 1 + x^3 \leftrightarrow 100100$ είναι

$$= \{000000, 100100, 010010, 001001, 110110, 101101, 011011, 111111\}.$$

Αυτό μπορούμε να το επιβεβαιώσουμε με τις τεχνικές που περιγράψαμε πρωτύτερα σ' αυτήν την ενότητα. Το πολυώνυμο ελάχιστου βαθμού που αντιπροπωπεύει μία λέξη του C είναι όπως παρατηρούμε το $g(x) = 1 + x^3$ και ο C δεν περιέχει κανένα άλλο πολυώνυμο βαθμού 3. Έτσι το $g(x) = 1 + x^3$ είναι το πολυώνυμο -γεννήτορας του C. Αναπαριστούμε κάθε λέξη του C ως πολλαπλάσιο του $g(x)$, (όπως φαίνεται στον παρακάτω πίνακα).

λέξη	πολυώνυμο $f(x)$	παραγοντοποίηση $h(x)g(x)$ του $f(x)$
000000	0	$0(1 + x^3)$
100100	$1 + x^3$	$1(1 + x^3)$
010010	$x + x^4$	$x(1 + x^3)$
001001	$x^2 + x^5$	$x^2(1 + x^3)$
110110	$1 + x + x^3 + x^4$	$(1 + x)(1 + x^3)$
101101	$1 + x^2 + x^3 + x^5$	$(1 + x^2)(1 + x^3)$
011011	$x + x^2 + x^4 + x^5$	$(x + x^2)(1 + x^3)$
111111	$1 + x + x^2 + x^3 + x^4 + x^5$	$(1 + x + x^2)(1 + x^3)$

Μπορούμε να δημιουργήσουμε κυκλικούς κώδικες πολύ εύκολα, επιλέγοντας μια λέξη v και θέτοντας $C = \langle\{v(x), xv(x), ..., x^{n-1}v(x)\}\rangle$ (modulo $1 + x^n$). Εντούτοις, πρέπει να βρούμε το γεννήτορα πολυώνυμο για έναν τέτοιο κώδικα και η καταγραφή όλων των κωδικολέξεων δεν είναι ο φυσιολογικότερος τρόπος. Το πολυώνυμο -γεννήτορας για ένα κυκλικό κώδικα έχει μία σπουδαία ιδιότητα:

Θεώρημα 4.2.17 *Το $g(x)$ είναι το πολυώνυμο -γεννήτορας για ένα γραμμικό κυκλικό κώδικα μήκους n εάν και μόνο εάν το $g(x)$ διαιρεί το $1 + x^n$ (άρα $1 + x^n = h(x)g(x)$).*

Απόδειξη: Από τον Αλγόριθμο της Διαίρεσης $1+x^n = h(x)g(x)+r(x)$ με $r(x) = 0$ ή ο βαθμός του $(r(x))$ < του βαθμού του $(g(x))$. Ισοδύναμα $r(x) = h(x)g(x)+(1+x^n)$. Όμως $r(x) = (h(x)g(x)+(1+x^n)) \bmod (1+x^n) = h(x)g(x) \pmod{1+x^n}$. Έτσι το $r(x)$ ανήκει στον κώδικα που παράγεται από το $g(x)$ και $r(x) = 0$ ή ο βαθμός του $(r(x)) \leq$ του βαθμού του $(g(x))$. Συμπεραίνουμε ότι $r(x) = 0$.

Πόρισμα 4.2.18 *Το πολυώνυμο -γεννήτορας $g(x)$ για τον μικρότερο κυκλικό κώδικα μήκους n που περιέχει τη λέξη v (πολυώνυμο $v(x)$) είναι ο μέγιστος κοινός διαιρέτης του $v(x)$ και του $1 + x^n$ (δηλαδή, $g(x) =$Μ.Κ.Δ.$(v(x), 1 + x^n)$).*

Απόδειξη: Εάν το $g(x)$ είναι το πολυώνυμο -γεννήτορας, τότε το $g(x)$ διαιρεί και το $v(x)$ και το $1 + x^n$. Όμως το $g(x)$ ανήκει στο $\langle\{v(x), xv(x), ..., x^{n-1}v(x)\}\rangle$, έτσι έχουμε

$$g(x) = a(x)v(x) \bmod 1 + x^n$$

ή ισοδύναμα με τον Αλγόριθμο της Διαίρεσης:

$$g(x) = a(x)v(x) + b(x)(1 + x^n).$$

Οπότε κάθε κοινός διαιρέτης των $v(x)$ και $1 + x^n$ πρέπει να διαιρεί το $g(x)$ και έτσι το $g(x)$ είναι ο μέγιστος κοινός διαιρέτης.

Παράδειγμα 4.2.19 Έστω $n = 8$ και $v = 11011000$, δηλαδή $v(x) = 1 + x + x^3 + x^4$. Ο Μ.Κ.Δ των $v(x)$ και $1 + x^8$ είναι το $1 + x^2$. Έτσι $g(x) = 1 + x^2$ και ο μικρότερος γραμμικός κυκλικός κώδικας που περιέχει το $v(x)$ έχει διάσταση 6 και το $g(x)$ είναι το πολυώνυμο -γεννήτορας.

Ο Αλγόριθμος του Ευκλείδη για τον υπολογισμό του Μ.Κ.Δ δύο πολυωνύμων συζητιέται στο παράρτημα Α. Μία άλλη προσέγγιση, για να βρούμε το γεννήτορα πολυώνυμο ενός κυκλικού κώδικα μήκους n και διάστασης $n - k$, χρησιμοποιεί απλή row reduction. Εάν πάρουμε μία βάση (ή έναν γεννήτορα πίνακα) και τη θέσουμε σε μορφή ΑΓΚΜ με τις τελευταίες k στήλες να είναι οι «στήλες οδηγοί» τότε η γραμμή (κωδικολέξη) ελαχίστου βαθμού θα είναι το πολυώνυμο -γεννήτορας.

Ασκήσεις

4.2.20 Για κάθε μία από τις παρακάτω λέξεις βρείτε το γεννήτορα πολυώνυμο για τον μικρότερο γραμμικό κυκλικό κώδικα που περιέχει τη λέξη αυτή.

(α΄). 010101

(β΄). 010010

(γ΄). 01100110

(δ΄). 0101100

(ε΄). 001000101110000

(ϛ΄). 000010010000000

(ζ΄). 010111010000000

4.2.21 Βρείτε το γεννήτορα πολυώνυμο του μικρότερου γραμμικού κυκλικού κώδικα που περιέχει κάθε μία από τις παρακάτω λέξεις.

(α΄). 101010

(β΄). 1100

(γ΄). 10001000

(δ΄). 011011

(ε΄). 10101

(ϛ΄). 111111.

4.2.22 Για κάθε έναν από τους κώδικες $C = \langle S \rangle$ με S να ορίζεται παρακάτω, βρείτε το γεννήτορα πολυώνυμο $g(x)$ και στη συνέχεια αναπαραστήστε κάθε λέξη του κώδικα ως πολλαπλάσιο του $g(x)$.

(α΄). $S = \{1010, 011, 111\}$

(β΄). $S = \{1010, 0101, 1111\}$

(γ΄). $S = \{0101, 1010, 1100\}$

(δ΄). $S = \{1000, 0100, 0010, 0001\}$

(ε΄). $S = \{111000, 01111, 11110, 01010\}$

4.3 Πολυωνυμική Κωδικοποίηση και Αποκωδικοποίηση

Μπορούμε να βρούμε πολλούς πίνακες γεννήτορες για γραμμικούς κυκλικούς κώδικες· ο πιο απλός είναι ο πίνακας στον οποίο οι γραμμές είναι οι κωδικολέξεις που αντιστοιχούν στο γεννήτορα πολυώνυμο και στις πρώτες $k-1$ κυκλικές

μεταθέσεις του (δείτε θεώρημα 4.2.13):

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

Παράδειγμα 4.3.1 Έστω $C = \{0000, 1010, 0101, 1111\}$ είναι ένας γραμμικός κυκλικός κώδικας. Το πολυώνυμο -γεννήτορας για τον C είναι το $g(x) = 1 + x^2$. Εδώ $n = 4$ και $k = 2$, οπότε μία βάση για τον C αποτελείται από τα

$$g(x) = 1 + x^2 \leftrightarrow 1010, xg(x) = x + x^3 \leftrightarrow 0101,$$

όπως εύκολα διαπιστώνουμε. Ένας γεννήτορας πίνακας για τον C είναι

$$G = \begin{bmatrix} g(x) \\ xg(x) \end{bmatrix} = \begin{bmatrix} 1010 \\ 0101 \end{bmatrix}.$$

Παράδειγμα 4.3.2 Έστω C είναι ο γραμμικός κυκλικός κώδικας μήκους $n = 7$ με γεννήτορα πολυώνυμο $g(x) = 1 + x + x^3$ βαθμού $n - k = 3$. Τότε $k = 4$, οπότε μία βάση για τον C είναι,

$$\begin{array}{ll} g(x) & = 1 + x + x^3 \\ xg(x) & = x + x^2 + x^4 \\ x^2g(x) & = x^2 + x^3 + x^5 \\ x^3g(x) & = x^3 + x^4 + x^6 \end{array}$$

και ένας γεννήτορας για τον C είναι

$$G = \begin{bmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0001101 \end{bmatrix}$$

Έστω C είναι ένας γραμμικός κυκλικός κώδικας μήκους n και διάστασης k (οπότε το πολυώνυμο -γεννήτορας $g(x)$ έχει βαθμό $n - k$). Τα k ψηφία που περιέχουν την πληροφορία $(a_0, a_1, ..., a_{k_1})$ και πρόκειται να κωδικοποιηθούν μπορούμε να τα σκεφτόμαστε ως το πολυώνυμο $a(x) = a_0 + a_1x + ... + a_{k-1}x^{k-1}$ το οποίο ονομάζεται *πολυώνυμο πληροφορίας* ή *πολυώνυμο μηνύματος*. Η κωδικοποίηση είναι ουσιαστικά πολλαπλασιασμός πολυωνύμων. Δηλαδή, το $a(x)$ κωδικοποιείται ως $a(x)g(x) = c(x)$. Οπότε αντί να αποθηκεύσουμε ολόκληρο τον $k \times n$ γεννήτορα πίνακα θα αποθηκεύσουμε μόνο το γεννήτορα πολυώνυμο, πράγμα που αποτελεί μία σημαντική βελτίωση ως προς την απλότητα της κωδικοποίησης.

Η αντίστροφη πράξη του πολυωνυμικού πολλαπλασιασμού είναι η πολυωνυμική διαίρεση. Έτσι η εύρεση του μηνύματος που αντιστοιχεί στην κοντινότερη κωδικολέξη $c(x)$ στην παραλειφθείσα λέξη πραγματοποιείται με τη διαίρεση του $c(x)$ με το $g(x)$, έτσι ώστε να αποκαλυφθεί το πολυώνυμο μηνύματος $a(x)$.

Παράδειγμα 4.3.3 Έστω $g(x) = 1 + x + x^3$ και $n = 7$. Τότε $k = 7 - 3 = 4$. Έστω $a(x) = 1 + x^2$ το πολυώνυμο μηνύματος που αντιστοιχεί στη λέξη $a = 1010$. Το μήνυμα $a(x)$ κωδικοποιείται με το $c(x) = a(x)g(x)$, οπότε

$$c(x) = (1 + x^2)(1 + x + x^3) = 1 + x + x^2 + x^5$$

με την $c = 1110010$ να είναι η αντίστοιχη κωδικολέξη.

Εάν $c(x) = 1 + x + x^4 + x^6$ τότε το αντίστοιχο πολυώνυμο μηνύματος είναι το $c(x)/g(x) = a(x) = 1 + x^3$ που αντιστοιχεί στο μήνυμα $a = 1001$.

Ασκήσεις

4.3.4 Έστω $g(x) = 1 + x + x^2$ είναι το γεννήτορα πολυώνυμο για γραμμικό κυκλικό κώδικα μήκους $n = 7$.

(α΄). Κωδικοποιείστε τα επόμενα πολυώνυμα μηνύματος: $1 + x^3$, x, $x + x^2 + x^3$.

(β΄). Βρείτε το πολυώνυμο μηνύματος που αντιστοιχεί στις κωδικολέξεις $c(x)$: $x^2 + x^4 + x^5$, $1 + x + x^2 + x^4$, $x^2 + x^3 + x^4 + x^6$.

4.3.5 Βρείτε μία βάση και έναν γεννήτορα πίνακα για το γραμμικό κυκλικό κώδικα μήκους n με γεννήτορα πολυώνυμο $g(x)$.

(α΄). $n = 7$, $g(x) = 1 + x^2 + x^3$

(β΄). $n = 9$, $g(x) = 1 + x^3 + x^6$

(γ΄). $n = 15$, $g(x) = 1 + x + x^4$

(δ΄). $n = 15$, $g(x) = 1 + x^4 + x^6 + x^7 + x^8$

(ε΄). $n = 15$, $g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$.

4.3.6 Δείξτε ότι ο γραμμικός κώδικας με το δοθέν γεννήτορα πίνακα είναι κυκλικός και βρείτε το γεννήτορα πολυώνυμο.

(α΄).

$$G = \begin{bmatrix} 110110 \\ 001001 \\ 101101 \\ 101101 \end{bmatrix}$$

(β΄).

$$G = \begin{bmatrix} 010101 \\ 111111 \end{bmatrix}$$

Έχοντας αναπτύξει έναν αλγόριθμο πολυωνυμικής κωδικοποίησης για γραμμικούς κυκλικούς κώδικες θα πρέπει στη συνέχεια να αναπτύξουμε έναν parity check πίνακα για τέτοιους κώδικες καθώς επίσης κάποιον αλγόριθμο για αποκωδικοποίηση των λαμβανόμενων λέξεων. Εάν το $c(x)$ έχει αποσταλεί και το $w(x)$ έχει παραληφθεί με $w(x) = c(x) + e(x)$ τότε θα θέλαμε να υπολογίσουμε το σύνδρομο και το πιο πιθανό πολυώνυμο λάθους $e(x)$.

Το *σύνδρομο πολυώνυμο,* $s(x)$, ορίζεται ως $s(x) = w(x) \bmod g(x)$. Υποθέτοντας ότι το $g(x)$ έχει βαθμό $n-k$, τότε το $s(x)$ θα έχει βαθμό μικρότερο από το $n-k$ και θα αντιστοιχεί σε μία δυαδική λέξη s, μήκους $n-k$. Επειδή $w(x) = c(x)+e(x)$ και $c(x) = a(x)g(x)$ παίρνουμε $s(x) = e(x) \bmod g(x)$. Δηλαδή το σύνδρομο πολυώνυμο εξαρτάται μονάχα από το σφάλμα.

Μπορούμε να ορίσουμε έναν πίνακα H στο οποίο η i-οστή γραμμή r_i είναι η λέξη μήκους $n-k$ που αντιστοιχεί στο $r_i(x) = x^i \bmod g(x)$. Αποδεικνύεται ότι ο πίνακας αυτός είναι ένας parity check πίνακας για τον κώδικα. Διότι εάν w είναι η παραληφθείσα λέξη τότε

$$\begin{aligned}
w(x) &= c(x) + e(x), \text{ έτσι} \\
wH &= (c+e)H \\
&= \sum_{i=0}^{n-1}(c_i + e_i)r_i \\
&\leftrightarrow \sum_{i=0}^{n-1}(c_i + e_i)r_i(x) \\
&= (\sum_{i=0}^{n-1} c_i x^i) \bmod g(x) + (\sum_{i=0}^{n-1} e_i x^i) \bmod g(x) \\
&= c(x) \bmod g(x) + e(x) \bmod g(x) \\
&= 0 + e(x) \bmod g(x) \\
&= s(x).
\end{aligned}$$

Τότε $s(x) = 0$ εάν και μόνο εάν το $w(x)$ είναι μια κωδικολέξη, οπότε ο H είναι ένας parity check πίνακας. Ακόμη εάν $wH = s$ τότε η s αντιστοιχεί στο $s(x) = w(x) \bmod g(x)$. Είναι τώρα φανερό γιατί λέμε το $s(x)$ σύνδρομο πολυώνυμο.

Παράδειγμα 4.3.7 Έστω $n = 7$, και $g(x) = 1+x+x^3$. Τότε $n-k = 3$. Παράγουμε τον H όπως ακολουθεί:

$$\begin{array}{lll}
r_0(x) = 1 & \bmod g(x) = 1 & \leftrightarrow 100 \\
r_1(x) = x & \bmod g(x) = x & \leftrightarrow 010 \\
r_2(x) = x^2 & \bmod g(x) = x^2 & \leftrightarrow 001 \\
r_3(x) = x^3 & \bmod g(x) = 1 + x & \leftrightarrow 110 \\
r_4(x) = x^4 & \bmod g(x) = x + x^2 & \leftrightarrow 011 \\
r_5(x) = x^5 & \bmod g(x) = 1 + x + x^2 & \leftrightarrow 111 \\
r_6(x) = x^6 & \bmod g(x) = 1 + x^2 & \leftrightarrow 101
\end{array}$$

$$\text{οπότε } H = \begin{bmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{bmatrix}.$$

Εάν το $w(x) = 1 + x^5 + x^6$ λαμβάνεται, τότε $w = 1000011$ και $wH = s = 110$ και $s(x) = 1 + x = 1 + x^5 + x^6 \bmod (1 + x + x^3)$.

Αντί να κατασκευάσουμε τη συνηθισμένη κανονική παράταξη αποκωδικοποίησης (ΚΠΑ) για έναν κυκλικό κώδικα, θα χρησιμοποιήσουμε ένα αλγόριθμο που εκμεταλεύεται τις συμμετρίες που ενυπάρχουν στους κυκλικούς κώδικες. Παρατηρήστε ότι εάν e είναι ο οδηγός συμπλόκου και εάν $s = eH$, τότε $s(x) = e(x) \bmod g(x)$ όπως δείχθηκε στη συζήτηση πριν στο παράδειγμα 4.3.7. Τότε $x^i s(x) \equiv x^i e(x) \bmod g(x)$ και έτσι τα σύνδρομα των κυκλικών μεταθέσεων του e είναι εύκολο να υπολογιστούν. Αντί να χρειάζεται να αποθηκεύσουμε μια ΚΠΑ χρησιμοποιούμε αυτήν την ιδιότητα.

Είναι σημαντικό να παρατηρήσουμε ότι εάν ο βαθμός του $e(x) <$ του βαθμού του $g(x)$ τότε $e(x) = e(x) \bmod g(x)$ και έτσι το σύνδρομο πολυώνυμο για το πολυώνυμο λάθους $e(x)$ είναι απλώς το $e(x)$ (δηλαδή $s(x) = e(x)$). Επίσης παρατηρούμε ότι εάν το $e(x)$ είναι ένας οδηγός συμπλόκου για έναν κυκλικό κώδικα μήκους n, τότε είναι και το $x^i e(x) \bmod 1 + x^n$.

Αλγόριθμος 4.3.8 (Για την αποκωδικοποίηση γραμμικών κυκλικών κωδίκων).

1. Υπολογίστε το σύνδρομο πολυώνυμο $s(x) = w(x) \bmod g(x)$, όπου w είναι η παραληφθείσα λέξη.

2. Για κάθε $i \geq 0$, υπολογίστε την $s_i \leftrightarrow s_i(x) = x^i s(x) \bmod g(x)$ (το σύνδρομο πολυώνυμο για την i-οστή κυκλική μετάθεση του w) μέχρι να βρεθεί το σύνδρομο s_j με $w_t(s_j) \leq t$. Τότε το πιο πιθανό πολυώνυμο λάθους είναι το $e(x) = x^{n-j} s_j(x) \bmod (1 + x^n)$.

Σημείωση Αυτός ο αλγόριθμος αποκωδικοποίησης θα διορθώνει μονάχα υποδείγματα λάθους $e(x)$ όπου για κάποιο i, $x^i e(x) \bmod (1+x^n)$ έχει βαθμό το πολύ $n - k$. Είναι πολύ πιθανό ότι υπάρχουν υποδείγματα λάθους με βάρος το πολύ t που δεν ικανοποιούν αυτήν την ιδιότητα. Αυτά τα υποδείγματα λάθους διορθώνονται από τον κώδικα όμως η κοντονότερη κωδικολέξη δε βρίσκεται από αυτόν τον αλγόριθμο. Όμως ο αλγόριθμος 4.3.8 θα χρησιμοποιηθεί στο κεφάλαιο 7 όταν θα συζητήσουμε για τα υποδείγματα εκρήξεων. Σ' αυτήν την περίπτωση, το ανάλογο του αλγορίθμου 4.3.8 θα δουλεύει πάντα.

Παράδειγμα 4.3.9 Έστω $n = 7$ και $g(x) = 1 + x + x^3$ είναι το πολυώνυμο - γεννήτορας για το γραμμικό κυκλικό κώδικα διόρθωσης ενός λάθους (ώστε $t = 1$). Εάν το $w(x) = x^2 + x^3$ παραλαμβάνεται τότε το $s(x) = w(x) \bmod g(x) = x^2 +$

$x^3 \bmod (1+x+x^3) = 1+x+x^2$ είναι το σύνδρομο πολυώνυμο. Στη συνέχεια υπολογίζουμε:

$$\begin{aligned} s_1(x) &= xs(x) \bmod g(x) = x(1+x+x^2) \bmod g(x) = 1+x^2 \\ s_2(x) &= x^2 s(x) \bmod g(x) = x(1+x^2) \bmod g(x) = 1, \end{aligned}$$

το οποίο έχει βάρος $1 \leq t$. Οπότε $j = 2$ και άρα

$$e(x) = x^{7-2} s2(x) \bmod (1+x^7) = x^5.$$

Οπότε το $c(x) = w(x) + e(x) = (x^2+x^3) + x^5$ είναι η πιο πιθανή κωδικολέξη.

Παράδειγμα 4.3.10 Έστω $n = 15$ και έστω $g(x) = 1+x^4+x^6+x^7+x^8$ είναι το πολυώνυμο -γεννήτορας για έναν κυκλικό κώδικα με $d = 5$. Έτσι όλα τα υποδείγματα λάθους με βάρος $t = 2$ ή λιγότερο διορθώνονται. Αποκωδικοποιούμε τη ληθφείσα λέξη $w = 110011100111000$. Εδώ $w(x) = l + x + x^4 + x^5 + x^6 + x^9 + x^{10} + x^{11}$.

Το σύνδρομο πολυώνυμο $s(x) = w(x) \bmod g(x)$ είναι

$$\begin{aligned} s(x) &= 1+x+x^3+x^4+x^5+x^6+x^7 \\ s_1(x) &= xs(x) = x+x^2+x^4+x^5+x^6+x^7+x^8 \bmod g(x) \\ &= 1+x+x^2+x^5 \\ s_2(x) &= x^2 s(x) \equiv x+x^2+x^3+x^6 \pmod{g(x)} \\ s_3(x) &= x^3 s(x) \equiv x^2+x^3+x^4+x^7 \pmod{g(x)} \\ s_4(x) &= x^4 s(x) \equiv 1+x^3+x^5+x^6+x^7 \pmod{g(x)} \\ s_5(x) &= x^5 s(x) \equiv 1+x \pmod{g(x)}, \text{ το οποίο έχει βάρος } 2 \leq t. \end{aligned}$$

Οπότε $e(x) = x^{15-5} s_5(x) \bmod (1+x^{15}) = x^{10}+x^{11}$. Οπότε

$$\begin{aligned} c(x) &= w(x) + e(x) \\ &= w(x) + (x^{10}+x^{11}) \\ &= 1+x+x^4+x^6+x^9 \end{aligned}$$

Ασκήσεις

4.3.11 Βρείτε έναν parity check πίνακα για το γραμμικό κυκλικό κώδικα μήκους 7 με γεννήτορα $g(x) = 1+x+x^2+x^4$.

4.3.12 Βρείτε έναν parity check πίνακα για έναν κυκλικό κώδικα μήκους n και με γεννήτορα $g(x)$:

(α'). $n = 6$, $g(x) = 1+x^2$

(β'). $n = 6$, $g(x) = 1+x^3$

(γ'). $n = 8$, $g(x) = 1+x^2$

(δ΄). $n = 9$, $g(x) = 1 + x^3 + x^6$

(ε΄). $n = 15$, $g(x) = 1 + x + x^4$ (παράγει έναν κώδικα Hamming)

(ϛ΄). $n = 23$, $g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$ (παράγει έναν κώδικα Golay)

(ζ΄). $n = 15$, $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ (παράγει έναν BCH 2- κώδικα διόρθωσης λαθών, που κατασκευάστηκε στο κεφάλαιο 5).

4.3.13 Το $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ παράγει ένα 2- γραμμικό κώδικα διόρθωσης λαθών μήκους 15. Χρησιμοποιώντας τον αλγόριθμο 4.3.8 αποκωδικοποιείστε τις παρακάτω ληφθείσες λέξεις οι οποίες κωδικοποιήθηκαν χρησιμοποιώντας τον C.

(α΄). 001000001110110

(β΄). 110001101000101

(γ΄). 001111101001001

(δ΄). 001000000110000

(ε΄). 110010000111010.

4.4 Βρίσκοντας Κυκλικούς Κώδικες

Για να κατασκευάσουμε ένα γραμμικό κυκλικό κώδικα μήκους n και διάστασης k, πρέπει να βρούμε ένα παράγοντα του $1 + x^n$ που έχει βαθμό $n - k$. Φυσικά ίσως υπάρχουν πολλές επιλογές ή ίσως καμία για δοθέντα n και k. Υπάρχει επίσης και το ερώτημα της ελάχιστης απόστασης για τους κυκλικούς κώδικες που δε μελετήσαμε, ένα ερώτημα που δεν το αντιμετωπίσαμε γενικά. Δε θα απασχοληθούμε με αυτό το θέμα εώς αργότερα.

Να επαναλάβουμε, το γεγονός ότι κάθε γεννήτορας πρέπει να διαιρεί το $1 + x^n$ μας δίνει τη δυνατότητα να βρούμε όλους τους γραμμικούς κυκλικούς κώδικες με κάποιο δοθέν μήκος n. Το μόνο που πρέπει να κάνουμε είναι να βρούμε όλους τους παράγοντες του $1 + x^n$, που σημαίνει ότι θα πρέπει πρώτα να βρούμε όλους τους ανάγωγους παράγοντες.

Ένα πολυώνυμο $f(x)$ στο $K[x]$ βαθμού τουλάχιστον ένα είναι *ανάγωγο* εάν δεν είναι το γινόμενο δύο πολυωνύμων του $K[x]$, τα οποία και τα δύο είναι βαθμού τουλάχιστον ένα. Να βρούμε τους ανάγωγους παράγοντες (οι οποίοι ουσιαστικά δίνουν όλους τους παράγοντες του $1 + x^n$) δεν είναι καθόλου εύκολο. Η παραγοντοποίηση του $1 + x^n$, $n \leq 31$ σε ανάγωγους παράγοντες βρίσκεται στο παράρτημα Γ και ένας αλγόριθμος για παραγοντοποίηση του $1 + x^n$ περιγράφεται στο παράρτημα Β.

Ο παράγοντας 1 του $1 + x^n$ έχει βαθμό 0 και έτσι παράγει έναν κυκλικό κώδικα διάστασης n. Αυτός ο κώδικας πρέπει να είναι ο K^n, το οποίο αποδεικνύει ότι ο K^n είναι κυκλικός. Μπορούμε επίσης, ως ειδική περίπτωση, να ορίζουμε τον

κώδικα {0} που αποτελείται μονάχα από την μηδενική λέξη μήκους n να είναι ο κυκλικός κώδικας με «γεννήτορα» το $g(x) = 0 = 1 + x^n \bmod 1 + x^n$.

Θα ονομάζουμε αυτούς τους γραμμικούς κυκλικούς κώδικες K^n και {0}, *μή γνήσιους κυκλικούς κώδικες*. Αλλιώς ο κώδικας είναι ένας *γνήσιος κυκλικός κώδικας*.

Παράδειγμα 4.4.1 Για $h = 3$, $1+x^3 = (1+x)(1+x+x^2)$ είναι η παραγοντοποίηση του $1 + x^3$ σε ανάγωγους παράγοντες. Έτσι υπάρχουν δύο γνήσιοι κυκλικοί κώδικες μήκους 3. Ο ένας έχει γεννήτορα $g(x) = 1 + x$ και γεννήτορα πίνακα

$$G = \begin{bmatrix} 110 \\ 011 \end{bmatrix}.$$

Ο κώδικας είναι ο $C = \{000, 110, 011, 101\}$. Ο άλλος κώδικας έχει γεννήτορα το $g(x) = 1 + x + x^2$ και γεννήτορα πίνακα $G = [111]$, οπότε είναι ο κώδικας $C = \{000, 111\}$.

Παράδειγμα 4.4.2 Για $n = 6$, παραγοντοποιούμε το $1 + x^6$ σε ανάγωγους παράγοντες.

$$1 + x^6 = (1 + x^3)^2 = (1 + x)^2(1 + x + x^2)^2.$$

Τότε για να βρούμε τους γεννήτορες των γνήσιων γραμμικών κυκλικών κωδίκων μήκους 6, σχηματίζουμε όλα τα πιθανά γινόμενα αυτών των παραγόντων εκτός του 1 και του $1 + x^6$. Κάθε ένα τέτοιο γινόμενο είναι ένας γεννήτορας για έναν γνλησιο κυκλικό κώδικα μήκους 6. Αυτά τα γινόμενα και η διάσταση του κυκλικού γραμμικού κώδικα μήκους 6 που κάθε γινόμενο παράγει δίνονται στον παρακάτω πίνακα.

γεννήτορας	διάσταση
$1 + x$	5
$(1 + x)^2 = 1 + x^2$	4
$1 + x + x^2$	4
$(l + x + x^2)^2 = 1 + x^2 + x^4$	2
$(1 + x)(1 + x + x^2) = 1 + x^3$	3
$(1 + x)^2(1 + x + x^2) = 1 + x + x^3 + x^4$	2
$(1 + x)(1 + x + x^2)^2 = 1 + x + x^2 + x^3 + x^4 + x^5$	1

Θεώρημα 4.4.3 *Εάν* $n = 2^r s$ *τότε* $1 + x^n = (1 + x^s)^{2^r}$.

Απόδειξη: Εάν $n = 2s$, τότε $(1+x^s)^2 = 1+x^s+x^s+x^{2s} = 1+x^{2s}$. Προχωρούμε με επαγωγή στο r.

Πόρισμα 4.4.4 *Έστω* $n = 2^r s$, *όπου* s *είναι περιττός και έστω* $1 + x^s$ *είναι το γινόμενο από* z *ανάγωγα πολυώνυμα. Τότε υπάρχουν* $(2^r + 1)^z$ *γραμμικοί κυκλικοί κώδικες μήκους* n *και* $(2^r + 1)^z - 2$ *γνήσιοι γραμμικοί κυκλικοί κώδικες μήκους* n.

Παράδειγμα 4.4.5 Στο παράδειγμα 4.4.1 δείχθηκε ότι το $1+x^3$ είναι το γινόμενο δύο ανάγωγων πολυωνύμων του $1+x$ και του $1+x+x^2$. Εφαρμόζοντας το πόρισμα 4.4.4 με $r = 0$, $s = 3$ και $z = 2$ βρίσκουμε ότι υπάρχουν $(2^0 + 1)^2 = 4$ γραμμικοί

κυκλικοί κώδικες μήκους 3, δύο εκ των οποίων είναι γνήσιοι (όπως δείχθηκε στο παράδειγμα 4.4.1). Επίσης, για το $1 + x^6$, έχουμε $n = 6 = 2^1 3$ οπότε $r = 1$, το z είναι ακόμη 2 έτσι υπάρχουν $(2 + 1)^2 = 9$ γραμμικοί κυκλικοί κώδικες μήκους 6, επτά εκ των οποίων είναι γνήσιοι (όπως δείχθηκε στο παράδειγμα 4.4.2).

Ασκήσεις

4.4.6 Βρείτε το πλήθος των γνήσιων κυκλικών κωδίκων μήκους n, όπου

(α'). $n = 4$,

(β'). $n = 5$,

(γ'). $n = 7$,

(δ'). $n = 14$,

(ε'). $n = 56$,

(ϛ'). $n = 15$,

(ζ'). $n = 120$,

(η'). $n = 1024$.

4.4.7 Βρείτε το γεννήτορα πολυώνυμο για όλους τους γνήσιους γραμμικούς κυκλικούς κώδικες μήκους n, όπου

(α'). $n = 4$,

(β'). $n = 5$.

4.4.8 Βρείτε δύο γεννήτορες βαθμού 4 για ένα γραμμικό κυκλικό κώδικα μήκους 7.

4.4.9 Βρείτε ένα γεννήτορα και ένα γεννήτορα πίνακα για ένα γραμμικό κώδικα μήκους n και διάστασης k όπου

(α'). $n = 12$, $k = 5$

(β'). $n = 12$, $k = 7$

(γ'). $n = 14$, $k = 5$

(δ'). $n = 14$, $k = 6$

(ε'). $n = 14$, $k = 8$.

4.4.10 Δείξτε ότι ο κώδικας Golay C_{23} είναι ισοδύναμος με ένα γραμμικό κυκλικό κώδικα.

Μπορούμε να βρούμε όλους τους κυκλικούς κώδικες ή ισοδύναμα να παραγοντοποιήσουμε το $1 + x^n$, με μία σχετικά απλή διαδικασία. Σε όλη τη μελέτη μας θα υποθέσουμε ότι το n είναι περιττός.

Το πρώτο βήμα περιλαμβάνει την παραγωγή όλων των πολυωνύμων $I(x) \pmod{1+x^n}$ έτσι ώστε $I(x) = I(x)^2 \pmod{1+x^n}$. Τέτοια πολυώνυμα ονομάζονται *αυτοδύναμα (idemponent)* πολυώνυμα. Είναι εύκολο να δούμε ότι εάν $u(x)$ και $v(x)$ είναι αυτοδύναμα, τότε είναι και το άθροισμα $u(x) + v(x)$ και το γινόμενο $u(x)v(x) \pmod{1+x^n}$. Έτσι είναι ανάγκη να κατασκευάσουμε μονάχα ένα «βασικό» σύνολο από αυτοδύναμα πολυώνυμα. Για να το κάνουμε αυτό χρειαζόμαστε να διαμερίσουμε το $Z_n = \{0, 1, ..., n-1\}$ σε «κλάσεις».

Έστω $C_i = \{s = 2^j \cdot i \pmod{n} | j = 0, 1...r\}$ όπου $1 = 2^r \bmod n$.

Παράδειγμα 4.4.11 Για $n = 7$ έχουμε

$$C_0 = \{0\}, C_1 = \{1, 2, 4\} = C_2 = C_4 \text{ και } C_3 = \{3, 5, 6\} = C_5 = C_7.$$

Για $n = 9$ έχουμε

$$C_0 = \{0\}, C_1 = \{1, 2, 4, 8, 7, 5\} \text{ και } C_3 = \{3, 6\}.$$

Στη συνέχεια για κάθε διαφορετική κλάση C_i σχηματίζουμε το πολυώνυμο

$$c_i(x) = \sum_{j \in C_i} x^j.$$

Ισχυριζόμαστε ότι το $c_i(x)$ είναι ένα αυτοδύναμο πολυώνυμο και επίσης ότι κάθε αυτοδύναμο πολυώνυμο $I(x) pmod 1 + x^n$ είναι ίσο με

$$I(x) = \sum_{i=0}^{k} a_i c_i(x), a_i \in \{0, 1\}.$$

Για να το δούμε αυτό, ας παρατηρήσουμε ότι,

$$c_i(x)^2 = c_i(x^2) = \sum_{j \in C_i} x^{2j} = \sum_{k \in C_i} x^k \pmod{1 + x^n}$$

διότι εάν το $j \in C_i$ τότε ανήκει και το $2j \pmod{n}$.

Παράδειγμα 4.4.12 Για $n = 7$ έχουμε,

$$\begin{aligned} C_0 = \{0\}, \text{ οπότε} \quad & c_0(x) = x^0 = 1, \\ C_1 = \{1, 2, 4\}, \text{ οπότε} \quad & c_1(x) = x^1 + x^2 + x^4 \text{ και} \\ C_3 = \{3, 5, 6\}, \text{ οπότε} \quad & c_3(x) = x^3 + x^6 + x^5. \end{aligned}$$

Τότε κάθε αυτοδύναμο πολυώνυμο $\pmod{1 + x^7}$ μπορεί να εκφραστεί ως

$$I(x) = a_0 c_0(x) + a_1 c_1(x) + a_3 c_3(x), a_i \in \{0, 1\}.$$

Έτσι έχουμε $2^3 - 1$ διαφορετικά αυτοδύναμα πολυώνυμα $\pmod{1 + x^n}$. (Δε θεωρούμε το $I(x) = 0$ το οποίο είναι ένα τετριμμένο αυτοδύμαμο).

Η σχέση μεταξύ των αυτοδύναμων πολυωνύμων και των κυκλικών κωδίκων είναι η εξής:

Θεώρημα 4.4.13 *Κάθε κυκλικός κώδικας περιέχει ένα μοναδικό αυτοδύναμο πολυώνυμο που παράγει του κώδικα.*

Απόδειξη: Έστω $g(x)$ είναι ο γεννήτορας ενός κυκλικού κώδικα μήκους n και έστω $g(x)h(x) = 1 + x^n$ (n είναι περιττός). Τότε ο Μ.Κ.Δ. $(h(x), g(x)) = 1$ και από τον Αλγόριθμο του Ευκλείδη (παράρτημα Α) υπάρχουν πολυώνυμα $t(x)$, $s(x)$ έτσι ώστε

$$1 = t(x)g(x) + s(x)h(x).$$

Πολλαπλασιάζοντας και τις δύο πλευρές με $t(x)g(x)$ παίρνουμε,

$$t(x)g(x) = (t(x)g(x))^2 + t(x)s(x)(1 + x^n)$$

ή

$$t(x)g(x) = (t(x)g(x))^2 \bmod 1 + x^n.$$

Έτσι το $t(x)g(x)$ είναι αυτοδύναμα και

$$g(x) = \text{Μ.Κ.Δ.}(t(x)g(x), 1 + x^n).$$

Παράδειγμα 4.4.14 Για να βρούμε όλους τους κυκλικούς κώδικες μήκους 9, βρίσκουμε όλα τα αυτοδύναμα πολυώνυμα και όλα τα αντίστοιχά τους πολυώνυμα γεννήτορες. Επειδή

$$C_0 = \{0\}, C_1 = \{1, 2, 4, 8, 7, 5\}, C_3 = \{3, 6\}$$

έχουμε

$$c_0(x) = 1, c_1(x) = x + x^2 + x^4 + x^5 + x^7 + x^8, c_3(x) = x^3 + x^6$$

και

$$I(x) = a_0c_0(x) + a_1c_1(x) + a_3c_3(x).$$

Αυτοπαθή πολυώνυμο $I(x)$	Το πολυώνυμο -γεννήτορας $g(x) = \text{Μ.Κ.Δ.}(I(x), 1 + x^9)$
1	1
$x + x^2 + x^4 + x^5 + x^7 + x^8$	$1 + x + x^3 + x^4 + x^6 + x^7$
$x^3 + x^6$	$1 + x^3$
$1 + x + x^2 + x^4 + x^5 + x^7 + x^8$	$1 + x + x^2$
$1 + x^3 + x^6$	$1 + x^3 + x^6$
$x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$	$1 + x$
$1 + x + x^2 + x^3 + x^5 + x^6 + x^7 + x^8$	$1 + x + x^2 + x^3 + x^5 + x^6 + x^7 + x^8$

Ασκήσεις

4.4.15 Βρείτε όλα τα αυτοδύναμα πολυώνυμα $\mathrm{mod} 1 + x^n$ και τα αντίστοιχα πολυώνυμα γεννήτορες για,

(α'). $n = 5$

(β'). $n = 7$

(γ'). $n = 11$

(δ'). $n = 15$

(ε'). $n = 31$

4.5 Δυϊκοί Κυκλικοί Κώδικες

Ένα άλλο γεγονός στους κυκλικούς κώδικες το οποίο είναι χρήσιμο, είναι ότι οι δυϊκοί κώδικες είναι επίσης κυκλικοί. Πράγματι θα δώσουμε μια διαδικασία για την κατασκευή του πολυωνύμου γεννήτορα του δυϊκού κώδικα.

Είναι απλό να δούμε ότι ο δυϊκός ενός κυκλικού κώδικα είναι και αυτός κυκλικός. Αυτό βγαίνει απ' ευθείας από το γεγονός ότι εάν $a \cdot b = 0$ τότε $\pi(a) \cdot \pi(b) = 0$ όπου π είναι η κυκλική μετάθεση, όπως η επόμενη απόδειξη δείχνει. (Σημειώστε ότι $a \cdot b = a_0b_0 + a_1b_1 + ... + a_nb_n$ και $\pi(a) \cdot \pi(b) = a_1b_1 + a_2b_2 + ... + a_nb_n + a_0b_0 = a \cdot b = 0$). Θεωρούμε τον κυκλικό κώδικα που παράγεται από τη λέξη v. Οπότε $C = \langle \{v, \pi(v), ..., \pi^{n-1}(v)\} \rangle$. Εάν $u \in C^\perp$ τότε $\pi^i(v) \cdot u = 0$ για $i = 0, 1, ..., n-1$. Όμως αυτό δείχνει ότι $\pi^{i+1}(v) \cdot \pi(u) = 0$ και ότι το $\pi(u)$ είναι ορθογώνιο στο $\langle \{\pi(v), \pi^2(v), ..., \pi^n(v)\} \rangle = C$ επειδή $\pi^n(v) = v$. Επειδή το $u \in C^\perp$ συνεπάγεται ότι $\pi(u) \in C^\perp$ συμπεραίνουμε ότι ο $C^\perp$ είναι κυκλικός.

Για να βρούμε το γεννήτορα του δυϊκού χρειαζόμαστε να σχετίσουμε το γινόμενο των πολυωνύμων και το εσωτερικό γινόμενο των διανυσμάτων.

Λήμμα 4.5.1 *Έστω $a \leftrightarrow a(x)$, $b \leftrightarrow b(x)$ και $b' \leftrightarrow b'(x) = x^n b(x^{-1}) \bmod 1 + x^n$, τότε $a(x)b(x) \bmod 1 + x^n = 0$ εάν και μόνο εάν $\pi^k(a) \cdot b' = 0$ για $k = 0, 1, ..., n-1$.*

Απόδειξη: Έστω $c(x) = a(x)b(x) \bmod 1 + x^n$. Τότε ο συντελεστής του x^k στο $c(x)$ είναι

$$c_k = a_k b_0 + a_{k+1} b_{n-1} + ... + a_{n-1} b_{k+1} + a_0 b_k + ... + a_{k-1} b_1$$

διότι $x^k \equiv x^{n+k} \pmod{1 + x^n}$. Σημειώστε ότι εάν $a = (a_0, a_1, ..., a_{n-1})$ και $b = (b_0, b_1, ..., b_{n-1})$ τότε $b' = (b_0, b_{n-1}, b_{n-2}, ..., b_1)$ και έτσι $c_k = \pi^k(a) \cdot b'$. Έτσι $c_k = 0$ για $k = 0, 1, ..., n-1$ εάν και μόνο εάν $c(x) = 0 = a(x)b(x) \bmod 1 + x^n$.

Ξανά, έστω C είναι ένας γραμμικός κυκλικός κώδικας μήκους n και $g(x)$ είναι ένα πολυώνυμο -γεννήτορας για τον C. Ξέρουμε ότι το $g(x)$ διαιρεί το $1 + x^n$ και έτσι υπάρχει ένα μοναδικό πολυώνυμο $h(x)$, έτσι ώστε $1 + x^n = g(x)h(x)$. Από το λήμμα 4.5.1 ξέρουμε ότι το $x^n h(x^{-1})$ ανήκει στο $C^\perp$, αλλά θέλουμε να βρούμε το γεννήτορα του $C^\perp$.

Θεώρημα 4.5.2 *Έστω C είναι ένας γραμμικός κυκλικός κώδικας μήκους n και διάστασης k με γεννήτορα $g(x)$ και εάν $1+x^n = g(x)h(x)$ τότε ο $C^\perp$ είναι ένας κυκλικός κώδικας με διάσταση $n-k$ και γεννήτορα του $x^k h(x^{-1})$.*

Απόδειξη: Επειδή ο C έχει διάσταση k, το $g(x)$ έχει βαθμό $n-k$ και έτσι το $h(x)$ έχει βαθμό k. Επειδή

$$g(x)h(x) = 1 + x^n$$

έχουμε

$$g(x^{-1})h(x^{-1}) = 1 + (x^{-1})^n$$

και

$$\begin{array}{ll} x^n g(x^{-1})h(x^{-1}) & = x^n(1+x^{-n} \\ x^{n-k} g(x^{-1}) x^k h(x^{-1}) & = 1 + x^n. \end{array}$$

Έτσι το $x^k h(x^{-1})$ είναι ένας παράγοντας του $1+x^n$, που έχει βαθμό k και είναι το πολυώνυμο -γεννήτορας για το γραμμικό κυκλικό κώδικα, $C^\perp$ με διάσταση $n-k$ που περιέχει το $x^n h(x^{-1})$.

Παράδειγμα 4.5.3 Το $g(x) = 1+x+x^3$ είναι ο γεννήτορας ενός κυκλικού κώδικα μήκους 7 και διάστασης $k = 7-3 = 4$. Επειδή ο $g(x)$ είναι ένας παράγοντας του $1+x^7$ μπορούμε να βρούμε ένα $h(x)$ τέτοιο ώστε $1+x^7 = g(x)h(x)$ κάνοντας τη διαίρεση. Σ' αυτήν την περίπτωση $h(x) = 1+x+x^2+x^4$. Ο γεννήτορας για το $C^\perp$ είναι $g^\perp(x) = x^4 h(x^{-1}) = x^4(l + x^{-1} + x^{-2} + x^{-4}) = 1 + x^2 + x^3 + x^4$ που αντιστοιχεί στην $1011100 = w$. Προφανώς $g \cdot w = (11010000) \cdot (1011l00) = 0$ και $\pi^k(g) \cdot w = 0$ επίσης. Σημειώστε ότι $g^\perp(x) \neq h(x)$.

Παράδειγμα 4.5.4 Έστω $g(x) = 1+x+x^2$ είναι ο γεννήτορας για ένα γραμμικό κυκλικό κώδικα μήκους 6. Βρίσκουμε $h(x) = 1+x+x^3+x^4$ ικανοποιεί την ισότητα $g(x)h(x) = 1+x^6$. Οπότε το $g^\perp(x) = x^4 h(x^{-1}) = x^4(1+x^{-1}+x^{-3}+x^{-4}) = x^4+x^3+x+1$ είναι ο γεννήτορας για το δυϊκό κώδικα. Σημειώστε ότι σ' αυτό το παράδειγμα $g^\perp(x) = h(x)$.

Ασκήσεις

4.5.5 Βρείτε το γεννήτορα πολυώνυμο για το δυϊκό κώδικα του κυκλικού κώδικα μήκους n που έχει γεννήτορα πολυώνυμο το $g(x)$ όπου:

(α'). $n = 6$, $g(x) = 1+x^2$

(β'). $n = 6$, $g(x) = 1+x^3$

(γ'). $n = 8$, $g(x) = 1+x^2$

(δ'). $n = 9$, $g(x) = 1+x^3+x^6$

(ε'). $n = 15$, $g(x) = 1+x+x^4$

(ϛ'). $n = 15$, $g(x) = 1+x^4+x^6+x^7+x^8$

(ζ'). $n = 23$, $g(x) = 1+x+x^5+x^6+x^7+x^9+x^{11}$

(η'). $n = 7$, $g(x) = 1+x+x^2+x^4$