ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΩΡΙΑ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΙ ΚΩΔΙΚΩΝ
ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ

Ημερομηνία παράδοσης	28/7/2017	
Ονόματα φοιτητών – Αρ. μητρώου	Κωνσταντίνος-Σπυρίδων Μώκος	Π15098
	Γιάννης Ψυχάρης	Π15187
	Αλέξανδρος Αλυκιώτης	Π15018

Πινακας Περιεχομένων

REED-MULLER CODE SIMULATOR

Γενικές πληροφορίες

Η εργασία αποτελείται από 2 αρχεία (client.py, server.py), τα οποια προσομοιώνουν την διαδικασία αποστολής δεδομένων με την χρήση κωδικοποίησης Reed Muller. Ο σκοπός χρήσης του Reed Muller είναι η δυνατότητα αποσφαλμάτωσης του μηνύματος που αποστέλλεται μετάξι των 2 εφαρμογών, εφόσον η αρχική λέξη δεν έχει αλλοιωθεί πολύ. Η διαδικασία αποσφαλμάτωσης προσομοιώνεται επίσης μέσω τον εφαρμογών.

Η εργασία έχει υλοποιηθεί σε γλώσσα προγραμματισμού python 2.7.10 και έχει γίνει χρήση βιβλιοθηκών της Sage.

Εκτέλεση προγράμματος

Αρχικά εκτελούμε το αρχείο server.py και στην συνεχεια το αρχείο client.py. Με την εκτέλεση του αρχείου client.py ο χρηστης καλείται να δώσει την ταξη του κωδικα (r) και το μηκος του κωδικα (m), ώστε να δημιουργηθεί ο αντίστοιχος κώδικας Reed Muller (RM) και στην συνεχεια το βάρος του λάθους που επιθυμεί (μέσα στο επιτρεπτό όριο) και το πλήθος τον λέξεων που θα έχουν λάθος.

Στην συνεχεια δημιουργούνται 20 τυχαίες λέξεις του κώδικα και προστίθεται τυχαίο λάθος βάρους που έχει ορίσει ο χρηστης και αποστέλλονται μαζί με τα ορίσματα για τον RM (r και m) με την χρήση socket στην εφαρμογή server.py

Στην εφαρμογή server.py γίνετε δημιουργία του κώδικα reed muller μέσω τον ορισμάτων που λαμβάνει από την εφαρμογή client.py και στην συνεχεια προσπαθεί να αποκωδικοποιήσει τις λέξεις. Στην περίπτωση που αποκωδικοποιηθεί επιτυχώς η λέξη εμφανίζει την αποκωδικοποιημένη λέξη, αλλιώς εμφανίζει μήνυμα σφάλματος.

Maximum_error_weight

Το μέγιστο βάρος λάθους που μπορεί ο κώδικας να εντοπίσει πρέπει να είναι μικρότερο από το μισό της ελάχιστης απόστασης των κωδικολέξεων του κώδικα (RM.minimum_distance()/2) και από το μέγιστο όριο λαθών που μπορεί να αποκωδικοποιήσει ο κώδικας (RM.decoder().decoding_radius()). Οποτε γίνετε χρήση του τύπου:

max_errors = min(RM.decoder().decoding_radius(), RM.minimum_distance()/2) - 1

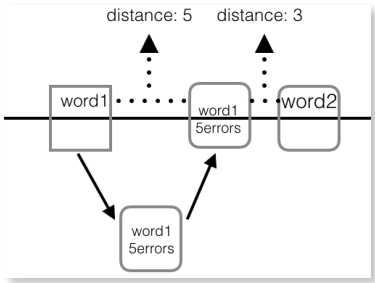
Και όπως παρατηρούμε με την χρήση τις μεθόδου: decoder_type() ο κώδικας είναι σε θέση να αποκωδικοποιεί πάντα τα λάθη μέχρι και το max_errors.

```
================================
========= Reed Muller =========
======= Code simulator ========
============ v0.1 =============
================================

Give reed-muller's order: 1
Give reed-muller's code length: 4
set(['bounded_distance', 'unique', 'always-succeed', 'hard-decision'])
Syndrome decoder for Binary Reed-Muller Code of order 1 and number of variables
4 handling errors of weight up to 3
** minimum distance: 8
```

Εικόνα 1:
Παράδειγμα για τον κώδικα RM(1,4) όπου ο decoder μπορεί να διορθώσει μέχρι και 3 λάθη όπως παρατηρούμε αφού εμφανίζεται: 'always-succeed'

Το μέγιστο βάρος που μπορεί να δώσει ο χρήστης εξαρτάται από τον αντίστοιχο κώδικα RM που δημιουργείται και είναι το μισό τις απόστασης των κωδικολεξεων (RM.minimum_distance()/2), καθώς εάν το βάρος του λάθους ξεπεράσει το όριο τότε υπάρχει πιθανότητα η τροποποιημένη λέξη να έχει μικρότερη απόσταση με κάποια άλλη λέξη μέσα στον κώδικα, δηλαδή υπάρχει περίπτωση ο κώδικας να μη αναγνωρίσει το σφάλμα σωστά.



Εικόνα 1:
Παράδειγμα για τον κώδικα RM(1,4) όπου ο η λέξη word1 έχει 5 λάθη με πιθανότητα ο κώδικας να νομίζει ότι η αρχική λέξη ήταν η word2

Αποστολή δεδομένων

Οι κωδικοποιημένες λέξεις μαζί με τις παραμέτρους για τη αρχικοποίηση του κώδικα RM αποστέλλονται στην εφαρμογή server.py αφού πρώτα προστεθούν σε μια λίστα (θέσεις: 0-19: encoded words | 20: r | 21: m) και γίνει serialization (pickle.dumps()).

Μόλις τα αρχεία φτάσουν στο server.py και αφού γίνει de-serialization (pickle.loads()) το πρόγραμμα επεξεργάζεται κατάλληλα τα δεδομένα.

Github Repo

Ο κώδικας της εργασίας υπάρχει επίσης ανεβασμένος στο github:
https://github.com/Psycharis/Reed-Mulller-code

Ατομικό Bonus

Κωνσταντίνος-Σπυρίδων Μώκος	(αποστολή με προσωπικό email)
Γιάννης Ψυχάρης	6972968657 - psycharisunipi@gmail.com
Αλέξανδρος αλυκιώτης	mkochila@gmail.com

Πηγές

Sage libraries	http://doc.sagemath.org/html/en/reference/coding/
python coding in general	https://stackoverflow.com/
sockets in python	https://docs.python.org/2/library/socket.html
Reed Muller theory	http://www-math.ucdenver.edu/~wcherowi/courses/m7823/reedmuller.pdf
Reed Muller theory	http://www.teilar.gr/dbData/ProfAnn/profann-ee2f9a9b.pdf
Python object serialization	https://docs.python.org/2/library/pickle.html