Formalization of Error-correcting Codes using SSReflect

Reynald Affeldt[1] and Jacques Garrigue[1]

[1]National Institute of Advanced Industrial Science and Technology
[2]Nagoya University

Abstract

By adding redundant information to transmitted data, error-correcting codes (ECCs) make it possible to communicate reliably over noisy channels. Minimizing redundancy and decoding time has driven much research, culminating with Low-Density Parity-Check (LDPC) codes. Hard-disk storage, wifi communications, mobile phones, etc.: most modern devices now rely on ECCs and in particular LDPC codes. Yet, correctness guarantees are only provided by research papers of ever-growing complexity. One solution to improve this situation is to enable certified implementations by providing a formalization of ECCs. We think that a first difficulty to achieve this goal has been partially lifted by the SSReflect [GMT08] library, that provides a substantial formalization of linear algebra. Using this library, we have been tackling the formalization of linear ECCs. In this talk, we would like to introduce a Coq library that makes it possible to formally state and verify basic properties of linear ECCs. This library is already rich enough to formalize most properties of Hamming codes and we are now in a position to formalize the more difficult LDPC codes.

1 Preliminaries about Linear ECCs

Let us first give a brief overview of the formal definitions we provide at the bottom layer of our library for ECCs. These formal definitions come as a complement to a previous formalization of information theory [AHS].

ECCs are about the manipulations of messages represented as bit-vectors. In SSReflect, bit-vectors are appropriately modeled as row vectors over the field $\mathbb{F}_2$ (type `'rV['F_2]_n`) and their properties are essentially discussed in terms of Hamming weight (the number of non-0 bits of a bit-vector) or Hamming distance.

The simplest definition of a linear ECC that one can find in the litterature is as a set of bit-vectors (the *codewords*) stable by addition. Yet, in practice, a linear code is rather defined as the kernel of a binary matrix called the *parity check matrix* [MS77]. The view of a linea ECC as a set makes it possible to prove several properties of ECCs such as the caracterization of the codewords of a given Hamming weight or the minimum Hamming distance between any two codewords.

Yet, the view of a linear ECC as a set abstracts aways the communication channel on which the transmission of codewords occur, as well as the coding procedure from messages to codewords, and the decoding procedure from transmitted codewords back to messages. From an operational viewpoint, a linear ECC is rather a pair of coding and decoding functions. The coding function turns a message into a codeword by cleverly adding redundancy. The resulting codeword is then sent into a communication channel modeled as a stochastic matrix. The decoding function recovers the original message from the output of the channel despite the noise. Using the view of a linear ECC as a pair of coding/decoding functions, it is possible to characterize and compare different encoding/decoding schemes according to wether they perform, e.g., minimum distance decoding (the decoding function decodes to the closest codeword in terms of Hamming distance) or maximum likelihood decoding (the decoding function decodes to the message that is the most likely to have been encoded according to the channel definition). Put formally, for an encoding function f, a maximum likelihood decoding function ϕ is such that:

$$W^n(y|f(\phi(y)) = \max_{x\in\mathbb{F}_2^n} W^n(y|f(x))$$

where $W^n(y|y_0)$ is the probability for a given channel that an input y_0 of length n is output as y.

2 Formalization of Hamming Codes

We used the formal definitions introduced in the previous section to formalize and prove the properties of Hamming codes, the most basic linear ECCs.

Hamming codes are linear codes on an alphabet Σ, where words of length $|\Sigma|^n - n - 1$ are encoded by codewords of length $|\Sigma|^n - 1$, *i.e.* one adds only n extra bits for error checking. Assuming binary words, the codewords y are defined by the parity check matrix, whose columns are the binary representations of non-null words of length n (there are $2^n - 1$ such words).

Let $[u]_i$ be the i^{th} bit of the word representing u in binary notation. Then codewords satisfy the equations:

$$\sum_{i=1}^{2^n-1} [i]_{j-1}[y]_{i-1} = 0 \ \ [\text{mod } 2] \quad (1 \le j \le n)$$

From the parity check matrix H, one can easily construct the encoding matrix G by permuting columns and using properties of block matrices.

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

We of course have the relation: $H \cdot G^T = 0$.

The most interesting property of Hamming codes is that there are no codewords of weights 1 and 2, *i.e.* by linearity the minimum distance between two distinct codewords is 3. A corollary of this property is that, by choosing the closest codeword, one is able to correct 1-bit errors (since there can be only one codeword that close).

Another, more abstract property is that the above approach of choosing the word corresponding to the closest codeword, *i.e.* the closest distance decoding, also happens to be a maximum likelihood decoding when the error probability of the channel is less than $\frac{1}{2}$.

We were able to prove those properties in the binary case, for arbitrary size source words, making efficient use of the matrix and finset libraries of SSReflect, and our own discrete probability framework. A large part of the effort was in finding the correct pre-conditions for each definition, something that is not insisted much on in textbooks. Note that, while computation on matrices can get tedious due to dependent types, we could avoid many problems by switching to sets of columns rather than concrete matrices where appropriate.

3 Towards Formalization of LDPC Codes

We are currently working, following [Hag12], on the formalization of LDPC codes and in particular the sum-product decoding algorithm. The sum-product algorithm is better explained using Tanner graphs as an alternative way to represent parity-check matrices. The figure on the right represents the Tanner graph corresponding to the Hamming code seen in Sect. 2. The sum-product algorithm decodes the outputs of a noisy channel by feeding each bit into the round nodes of the Tanner graph. The latter propagates bit information to connected square nodes where the received bits of information are used together with parity-check equations to estimate what information they should have received. This information is then sent back to round nodes to decide by majority vote whether or not the original bit they were given was wrong.

At the time of this writing, we have already formalized Tanner graphs, enough definitions to state a soundness property of the sum-product algorithm that we are in the process of formally proving.

Acknowledgements Kazuhiko Sakaguchi formalized Tanner graphs. Formalization of Hamming codes was done in collaboration with Taku Asai, Yūto Takahashi, and Takafumi Saikawa. This project is a collaboration with Manabu Hagiwara, Kenta Kasai, and Shigeaki Kuzuoka.

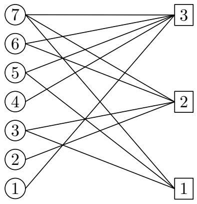
References

[AHS] Reynald Affeldt, Manabu Hagiwara, and Jonas Sénizergues. Formalization of shannon's theorems. *J. Autom. Reasoning*. To appear.

[GMT08] Georges Gonthier, Assia Mahboubi, and Enrico Tassi. A small scale reflection extension for the coq system. Technical Report RR-6455, INRIA, 2008. Version 14 (March 2014).

[Hag12] Manabu Hagiwara. *Coding Theory: Mathematics for Digital Communication*. Nippon Hyoron Sha, 2012. `http://www.nippyo.co.jp/book/5977.html`. In Japanese.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977. 7th impression: 1992.


7
6
5
4
3
2
1
3
2
1