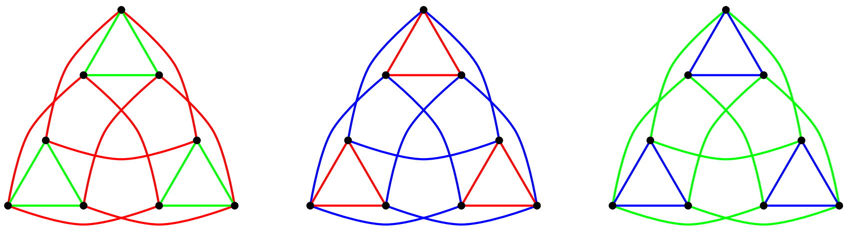
On the maximal size of codes of length 3 with the 2-Identifiable Parent Property

Mink Assies



On the maximal size of codes of length 3 with the 2-Identifiable Parent Property

Mink Assies

Master Thesis

August 16, 2004

Thesis supervisor: Dr. W. Bosma
Mathematical Department University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands

Preface

Imagine the following situation: a certain digital file contains information that is only to be available to a designated group of people and if the file is found outside this group, you would like to be able to find out who is guilty of passing it on.
Therefore each copy of the file has a personal codeword hidden in it. Now, if you find an illegal copy with someone's personal codeword in it, you will know who is responsible. This is called fingerprinting.
However, knowing this, several legal users could decide to cooperate to produce a copy with a new (code)word in it. They don't know how the code is hidden in their files, but by comparing their copies they can detect a number of positions in their files that are marked. They can alter the entries in those positions to produce a copy with a new (code)word in it. The more conspirators, the more positions of the code they potentially can detect and alter.
Given a maximum on the number of users that will conspire, you will want to create such a code that enables you to trace a newly created (code)word back to one of the guilty users.
A very simple method would be to assign a tuple to the ith user, with all entries zero, except on the ith position. However, this means that the length of the code grows linearly with the number of users and this is very impractical if you have millions of users. So you will want to keep the code as short as possible and still be able to handle enough users.
To be able to handle a certain number of users, it is possible to adjust the length of the code and work over an alphabet of fixed size. But I will follow a different approach: adjust the size of the alphabet and let the length of the code be fixed.
As a matter of fact, I will even do it the other way around: I will determine how many users I can serve for each size of the alphabet.

Enjoy.

Acknowledgments

First of all I would like to thank my supervisor Wieb Bosma without whom I would never have achieved the results of this thesis. I was very fortunate to have a supervisor who made so much time free for supervising. Furthermore, I enjoyed working with him socially; he is a very nice person with whom I could speak about other things than mathematics too. His knowledge of the English language was a great help too; it became very obvious to me that I had forgotten a thing or two about English since highschool.

Secondly, I would like to thank Dr. Ruud Jeurissen for being my second reader and Marcel Gorissen for being my unofficial second reader.

And last but not least, I would like to thank all the people who supported and motivated me. Quit a few of my fellow students and volleyball teammates read my thesis under construction and gave me valuable advise. Especially, I would like to thank Daan Wanrooy and Meron Brouwer for helping me (almost on a daily basis) with my LaTeX problems, Stijn Meurkens for writing the Magma function that was very useful when creating the table in chapter 7 and my parents and Mervijn Denie, a close friend, for motivating me to keep up the pace.

Contents

Chapter 1

Overview

In this thesis I will derive the upper bound $M(q) = \lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$ for the size of codes of length 3 with the 2-Identifiable Parent Property, given the size q of the alphabet.
For special values of q ($q = r^2 + 2r$, $r \in \mathbb{N}_{>0}$) a code of size $M(q)$ is constructed in the article of Hollmann et al. They constructed a code of size $3q - 6\sqrt{q+1} + 6$ for such special q and proved the general upper bound $3q - 1$ for the size of codes of length 3.
I will also suggest a construction for codes of size almost $3q - 6\sqrt{q+1} + 6$ for $q \neq r^2 + 2r$.
Essential in my proof are chapter 5, where I show that certain obstacles can be removed without difficulty, and theorem 3.2, which gives a relation between 2-IPP codes and their graphs. In chapter 7 I prove that $M(q)$ is an upper bound for the size of codes of length 3.

Chapter 2

Definitions and notation

1. A will denote the alphabet over which we will work; q will denote the size of A. I will generally assume A to be the set $\{1, \ldots, q\}$.

2. V will denote the set of all tuples of length l over A. An element of V shall be called a *word*.

3. Γ will denote a *code*. A code is a subset of V. An element of a code shall be called a *codeword*. From chapter 4 on I will only consider codes of length $l = 3$.
   Do not confuse the length of a code with the size of a code. The size is the number of codewords of which it consists and the length is the length of the codewords.

4. The *set of descendants of a subset W of V*, denoted by $D(W)$, is given by

   $D(W) = \{v \in V \mid \forall i \in \{1, \ldots, l\} : \; v_i \in \{w_i \mid w \in W\}\} =$
   $= W|_1 \times W|_2 \times \ldots \times W|_l$
   where $W|_i$ is the set of all symbols that occur on the ith coordinate in W.

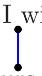
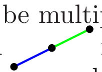
   This is the set of all tuples that can be obtained from the tuples in the subset W as follows: choose the symbol in the first position among the symbols in the first position of the tuples in the subset W, and so on for all positions.

5. A code Γ has the *c-Identifiable Parent Property*, shortly 'is a c-IPP code', if

$$\forall r \in \mathbb{N}_{>0} : \; D(C_1) \cap \ldots \cap D(C_r) \neq \emptyset \;\Rightarrow\; C_1 \cap \ldots \cap C_r \neq \emptyset$$

   where C_i is a set of at most c codewords.
   Such a C_i is called a *coalition*.
   This means that if several subsets can have the same tuple as a descendant, they must have at least one tuple in common. So this 'identifies' a parent.
   Examples of trivial cases are $r = 1$ or all C_i are of size 1.

6. To every code Γ we associate a graph with coloured edges, the *associated graph of* Γ , as follows:
   the codewords correspond bijectively to the vertices, and vertices are connected by an edge of colour i if and only if the corresponding codewords have the same symbol at the i-th coordinate.
   In our case, where we work with tuples of length 3, blue edges represent identical first coordinates, green edges represent identical second coordinates and red edges represent identical third coordinates.
   By a *(sub)graph of colour i* I mean a (sub)graph that contains edges of colour i only. By a (sub)graph of colour i and j I mean a (sub)graph that contains edges only of colour i and j. And so on.
   For sake of simplicity, I will, for instance, speak of 'coordinates of vertices'. Naturally, by this I will mean the coordinates of the tuple corresponding to the vertex.
   The code and its associated graph will sometimes be identified, so 'Γ' can denote a code and its associated graph.

7. By a *part* of a graph Γ I will mean a subgraph consisting of connected components of Γ (possibly including components of size 1).
   So a *part* does not have to be connected.
   Symbolically I will represent a part by its 'minimal graph'.
   For instance,  represents a part that contains at least one blue edge and no other colours. There can be large blue connected components, there can be multiple blue components and there can be isolated vertices.
   And  represents a part that contains at least one blue edge and one green edge and no red edges.
   These representations will really be used from chapter 6

8. A *partition of a graph* is a partition of the set of the connected components of the graph.

Chapter 3

Basic results on IPP codes

The following lemma gives an alternative characterization of 2-IPP.
IPP1 covers the situation that there is a set of three codewords of which all subsets of size 2 have a descendant in common.
IPP2 covers the situation that two disjoint sets of two codewords can have a descendant in common.
These are two of the most simple examples of non-2-IPP codes. It turns out that dealing with these two situations ensures 2-IPP.
Furthermore, this lemma inspires theorem 3.2.
The lemma is true for codes of all lengths, not just for $l = 3$.

Lemma 3.1 *A code Γ is a 2-IPP code if and only if it satisfies IPP1 and IPP2, where*
IPP1 means: $a, b, c \in \Gamma$ distinct $\Rightarrow \exists i : a_i, b_i, c_i$ distinct in A
IPP2 means: $a, b, c, d \in \Gamma : \{a, b\} \cap \{c, d\} = \emptyset \Rightarrow \exists i : \{a_i, b_i\} \cap \{c_i, d_i\} = \emptyset$

Proof:
Suppose a code Γ does not satisfy IPP1 and IPP2.
First suppose it does not satisfy IPP1.
Then $\exists a, b, c \in \Gamma$ *distinct* and $\forall i : a_i = b_i \lor a_i = c_i \lor b_i = c_i$.
This implies that there is a codeword d with $d \in D(\{a, b\}) \cap D(\{a, c\}) \cap D(\{b, c\})$ and $\{a, b\} \cap \{a, c\} \cap \{b, c\} = \emptyset$.
So Γ is not a 2-IPP code.
Now suppose Γ does not satisfy IPP2.
Then $\exists a, b, c, d \in \Gamma : \{a, b\} \cap \{c, d\} = \emptyset$ and $\forall i : a_i = c_i \lor a_i = d_i \lor b_i = c_i \lor b_i = d_i$.

This implies that there exists a word v with $v \in D(\{a,c\}) \cap D(\{b,d\})$ and $\{a,c\} \cap \{b,d\} = \emptyset$, or there exists a word v with $v \in D(\{a,d\}) \cap D(\{b,c\})$ and $\{a,d\} \cap \{b,c\} = \emptyset$.
So Γ is not a 2-IPP code.
Thus if Γ does not satisfy IPP1 and IPP2, Γ is not a 2-IPP code.

Suppose Γ is not a 2-IPP code.
Then $\exists\ r \in \mathbb{N}_{>0}, C_1, \ldots, C_r \subset \Gamma$ with $|C_i| \leq 2 :\ D(C_1) \cap \ldots \cap D(C_r) \neq \emptyset\ \wedge\ C_1 \cap \ldots \cap C_r = \emptyset$. Say $v \in D(C_1) \cap \ldots \cap D(C_r)$.
To avoid trivial cases I will assume $r > 1$, that there is at least one C_i with size 2 and that there are at least two disjoint coalitions.
Suppose there is only one C_i with size 2. Let $C_1 = \{x^{(1)}, y^{(1)}\}$ and $C_2 = \{x^{(2)}\}$ with $x^{(1)}, y^{(1)}, x^{(2)}$ different.
Then $\forall i :\ x_i^{(1)} = x_i^{(2)} \vee y_i^{(1)} = x_i^{(2)}$ because $v_i = x_i^{(1)} \vee v_i = y_i^{(1)}$ and $v_i = x_i^{(2)}$.
So Γ does not satisfy IPP1.
Suppose there are at least two C_i with size 2. Let $C_1 = \{x^{(1)}, y^{(1)}\}$ and $C_2 = \{x^{(2)}, y^{(2)}\}$.
Then $\forall i :\ \{x_i^{(1)}, y_i^{(1)}\} \cap \{x_i^{(2)}, y_i^{(2)}\} \neq \emptyset$ because $v_i \in \{x_i^{(1)}, y_i^{(1)}\}$ and $v_i \in \{x_i^{(2)}, y_i^{(2)}\}$.
So Γ does not satisfy IPP2.
Thus if Γ is not a 2-IPP code. Γ does not satisfy IPP1 and IPP2.

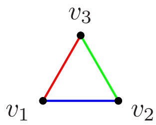
So a code Γ is a 2-IPP code if and only if it satisfies IPP1 and IPP2.
QED

From now on I will assume that codes have at least $q + 1$ elements, since fewer would be trivial: for a code of size $n \leq q$ you can take
$\Gamma = \{(0, 0, \ldots, 0), (1, 1, \ldots, 1) \ldots, (n-1, n-1, \ldots, n-1)\}$.
Such a code is always a c-IPP code (for any c).

Theorem 3.2 *A code of length 3 is a 2-IPP code if and only if the following three types of configuration do not occur in its associated graph:*

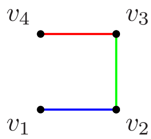
- *a 3-cycle of the three different colours;*
- *a path of length 3 of the three different colours;*
- *two edges of different colour between two vertices.*

Proof:
First I will prove that if Γ is a 2-IPP code of length 3 the three configurations cannot occur.
Let Γ be a 2-IPP code of length 3.

- suppose 

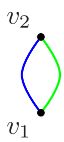
occurs.

  Then $v_1 = (a, x, c)$, $v_2 = (a, b, y)$ and $v_3 = (z, b, c)$ for some $a, b, c, x, y, z \in A$.
  Now, (a, b, c) is descendant of $\{v_1, v_2\}$, $\{v_1, v_3\}$ and $\{v_2, v_3\}$ but $\{v_1, v_2\} \cap \{v_1, v_2\} \cap \{v_1, v_2\} = \emptyset$.
  Thus Γ is not a 2-IPP code which is a contradiction.
  So this configuration cannot occur.
  Notice that this configuration conflicts with IPP1.

- suppose 

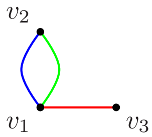
occurs.

  Then $v_1 = (a, p, r)$, $v_2 = (a, b, s)$, $v_3 = (x, b, c)$ and $v_4 = (y, z, c)$ for some $a, b, c, p, r, s, x, y, z \in A$.
  Now, (a, b, c) is descendant of $\{v_1, v_3\}$ and $\{v_2, v_4\}$ but $\{v_1, v_3\} \cap \{v_2, v_4\} = \emptyset$.
  Thus Γ is not a 2-IPP code which is a contradiction.
  So this configuration cannot occur.
  Notice that this configuration conflicts with IPP2.

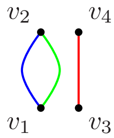
- suppose 

occurs.

  As stated below lemma 3.1 I assume that codes have at least $q + 1$ elements. That means that there is at least one edge of each colour.
  There are two possibilities for the red edge: it is incident with one of v_1 and v_2 or it is not.

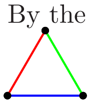
– suppose 

occurs. (The case where v_1 and v_2 are interchanged is similar.)
Then $v_1 = (a, b, c)$, $v_2 = (a, b, x)$ and $v_3 = (y, z, c)$ for some $a, b, c, x, y, z \in A$.
Now, (a, b, c) is descendant of $\{v_1\}$ and $\{v_2, v_3\}$ but $\{v_1\} \cap \{v_2, v_3\} = \emptyset$.
Thus Γ is not a 2-IPP code which is a contradiction.
So this configuration cannot occur.
Notice that this configuration conflicts with IPP1.

– suppose 

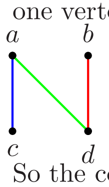
occurs.
Then $v_1 = (a, b, p)$, $v_2 = (a, b, r)$, $v_3 = (s, x, c)$ and $v_4 = (y, z, c)$ for some $a, b, c, p, r, s, x, y, z \in A$.
Now, (a, b, c) is descendant of $\{v_1, v_4\}$ and $\{v_2, v_3\}$ but $\{v_1, v_4\} \cap \{v_2, v_3\} = \emptyset$.
Thus Γ is not a 2-IPP code which is a contradiction.
So this configuration cannot occur.
Notice that this configuration conflicts with IPP2.

Now I will prove that if a code Γ is not a 2-IPP code, then one of the above mentioned configurations must occur.
I will prove this using lemma 3.1. So I will assume that Γ does not satisfy IPP1 or that Γ does not satisfy IPP2.
If the configuration of two edges of different colour between two vertices occurs I am done, so I may assume that it does not occur.
Suppose Γ does not satisfy IPP1:
$\exists a, b, c \textit{ distinct } : \forall i : a_i, b_i, c_i \textit{ not distinct}.$
So between the three vertices a, b and c there are three edges, each of one of the colours.
By the above assumption, there is essentially only one way to do this:



So the configuration of a 3-cycle of the three different colours occurs.
Suppose Γ does not satisfy IPP2:
$\exists a, b, c, d : \{a, b\} \cap \{c, d\} = \emptyset \;\wedge\; \forall i : \{a_i, b_i\} \cap \{c_i, d_i\} \neq \emptyset.$

Both sets $\{a, b\}$ and $\{c, d\}$ must be of size 2 or else there will be two edges of different colour between two vertices.
The only possibility for three edges between the two sets is that in both sets one vertex is connected by two edges and the other by one edge. For example:



So the configuration of a path of length 3 of the three different colours occurs.

So a code is a 2-IPP code of length 3 if and only if the three types of configuration do not occur in its associated graph.
QED

The configurations from this theorem shall be called the *forbidden configurations*.

The following lemma is true for codes of all lengths, not just for $l = 3$.

Lemma 3.3 *Let π be a permutation of A, let σ be a permutation of $\{1, \ldots, l\}$ and let Γ be a c-IPP code.*
Then $\Gamma' := \{(\pi(b_1), b_2, \ldots, b_l) | b \in \Gamma\}$ and $\Gamma'' := \{(b_{\sigma(1)}, \ldots, b_{\sigma(l)}) | b \in \Gamma\}$ are again c-IPP codes (over the same alphabet and of the same size).

Proof:
Let Γ be a c-IPP code and π a permutation of A.
Let $\Gamma' := \{(\pi(b_1), b_2, \ldots, b_l) | b \in \Gamma\}$.
It is obvious that $|\Gamma'| = |\Gamma|$ and that Γ' is defined over the same alphabet as Γ.
Suppose Γ' is not a c-IPP code.
Then there are coalitions ${C_1}', {C_2}', \ldots, {C_r}' \subset \Gamma'$, $|C_i'| \leq c$ with $D({C_1}') \cap \ldots \cap D({C_r}') \neq \emptyset$, say $a' \in D({C_1}') \cap \ldots \cap D({C_r}')$, and ${C_1}' \cap \ldots \cap {C_r}' = \emptyset$.
If I let the permutation $\rho = \pi^{-1}$ act on the first coordinates of Γ', and C_i' is mapped to C_i and a' to a, then $a \in D(C_1) \cap \ldots \cap D(C_r)$ and $C_1 \cap \ldots \cap C_r = \emptyset$.
This is a contradiction.
Hence if Γ is a c-IPP code, so is Γ' (and they are of the same size and defined over the same alphabet).

Similarly for Γ''.
QED

This lemma says that you can freely permute the coordinates of the code as a whole and you can freely permute the symbols on one of the coordinates of the code as a whole without losing the c-IPP. (Not only at the first coordinate: combine both statements of the lemma.)
This emphasizes the importance of the associated graph of a 2-IPP code.

Chapter 4

A lower bound code and an upper bound

In this chapter I will construct a code that shall provide a lower bound on the maximal size of a code of length 3, given the size of the alphabet q, and I shall prove the upper bound $3q - 1$ for the size of codes of length 3. Both results were given in the article of Hollmann et al.
Even though I clearly do not prove anything new in this chapter, it is very instructive.
In the following section for special q a code will be constructed that is of maximal size.

4.1 The lower bound code

Assume $q = r^2 + 2r$ (where r is a natural number).
Divide $A = \{1, 2, \ldots, r^2 + 2r\}$ into three disjoint classes:
$S := \{1, 2, \ldots, r\}$ (the *small* symbols)
$M := \{r + 1, r + 2, \ldots, 2r\}$ (the *medium* symbols)
$L := \{2r + 1, 2r + 2, \ldots, r^2 + 2r\}$ (the *large* symbols)
The code Γ will consist of the following three (disjoint) subsets:
$\Gamma_1 := \{(s_1, s_2, rs_1 + s_2 + r) | s_1, s_2 \in S\} \subseteq S \times S \times L$
$\Gamma_2 := \{(m, sr + m, s) | m \in M, s \in S\} \subseteq M \times L \times S$
$\Gamma_3 := \{(rm_1 + m_2 - r^2, m_1, m_2) | m_1, m_2 \in M\} \subseteq L \times M \times M$
Notice that $|S| = r$, $|M| = r$, $|L| = r^2$ and it is not difficult to verify that

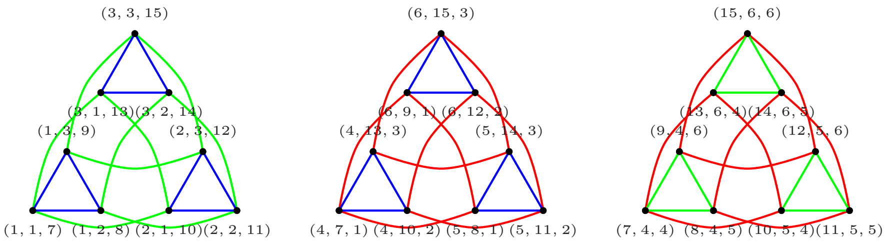
$\Gamma_1|_3 = L$, $\Gamma_2|_2 = L$ and $\Gamma_3|_1 = L$ and thus that $|\Gamma_1| = |\Gamma_2| = |\Gamma_3| = r^2$.

In the article of Hollmann et al it is shown that Γ is a 2-IPP code and that the size is $3r^2 = 3q - 6\sqrt{q+1} + 6$. (Furthermore they claim that for the general case, q does not have to be of a special form, one can get a lower bound of $3q - 12\sqrt{q}$ in a similar way).
However, they do not use graphs for this.

I would like to show that Γ is a 2-IPP code by considering its graph. Looking at the graphs of 2-IPP codes has inspired my results.
The first coordinates of the codewords in Γ_1 come from S, i.e. $\Gamma_1|_1 \subset S$. Likewise, $\Gamma_2|_1 \subset M$ and $\Gamma_3|_1 \subset L$.
Since S, M and L are disjoint there will be no blue edges between vertices of different subsets of Γ.
Similarly, there will be no edge of the other two colours between vertices of different subsets of Γ either.
So the partition of Γ in Γ_1, Γ_2 and Γ_3 induces a partition of its associated graph into three parts.
In each of the parts two colours are used, one is not.
For example, in the part of the graph belonging to Γ_1 two vertices can have the same first coordinate or the same second coordinate (not both at the same time, since then they would also have the same third coordinate and therefore be one and the same vertex) so the colours blue and green are used in this part.
However two vertices cannot have the same third coordinate:
Consider a codeword from Γ_1, $(s_1, s_2, rs_1 + s_2 + r)$.
If $s_1 = 1$ and $s_2 = 1 \ldots r$ then $rs_1 + s_2 + r = 2r + 1 \ldots 3r$
If $s_1 = 2$ and $s_2 = 1 \ldots r$ then $rs_1 + s_2 + r = 3r + 1 \ldots 4r$
$\vdots$
If $s_1 = r$ and $s_2 = 1 \ldots r$ then $rs_1 + s_2 + r = r^2 + r + 1 \ldots r^2 + 2r$
So each element of L is found exactly once. Therefore there are no red edges in this part of the graph.
Since in each part of the graph no two vertices can have the same symbol on two coordinates (no two edges of different colour between two vertices) and only two colours are used in each part (no 3-cycles of the three different colours or paths of length 3 of the three different colours), none of the forbidden configurations can occur and therefore Γ is a 2-IPP code.

An example for $q = 15$ ($r = 3$):

$$\begin{aligned}
\Gamma_1 &= \{(1,1,7),(1,2,8),(1,3,9),(2,1,10),(2,2,11),(2,3,12),(3,1,13),\\
&\quad (3,2,14),(3,3,15)\}\\
\Gamma_2 &= \{(4,7,1),(4,10,2),(4,13,3),(5,8,1),(5,11,2),(5,14,3),(6,9,1),\\
&\quad (6,12,2),(6,15,3)\}\\
\Gamma_3 &= \{(7,4,4),(8,4,5),(9,4,6),(10,5,4),(11,5,5),(12,5,6),(13,6,4),\\
&\quad (14,6,5),(15,6,6)\}
\end{aligned}$$



Looking at codes in this way gave me the idea that the optimal solution could be a code of which the graph consists of three parts, each missing one of the three colours (and every part a different one of course).

4.2 The upper bound $3q - 1$

In this section I will consider not the alphabet A for the code as a whole, but I will consider separate alphabets for each coordinate, denoted by (A_1, A_2, A_3), which may differ. So A will now be regarded as (A, A, A).
Consider the graph of a 2-IPP code Γ and pick a connected component P. (From now on, when I speak of a *component*, I mean a connected component.)
The alphabet can be devided in two: $({A_1}', {A_2}', {A_3}')$ for P and $({A_1}'', {A_2}'', {A_3}'')$ for the rest of the graph of Γ.
This is a partition of the alphabet on each coordinate since P is a connected component; if there would be, for instance, a vertex a outside P with the same symbol on the first coordinate as a vertex b in P they would be connected by a blue edge and a would actually be an element of P which is a contradiction, so no vertex outside P can use a symbol that is already used in P.

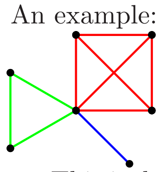
Just as q denotes the size of A, ${q_1}'$ denotes the size of ${A_1}'$, ${q_3}''$ denotes the size of ${A_3}''$ and so on.
As a convention $({A_1}', {A_2}', {A_3}')$ will not contain symbols that are not used. So ${A_i}'$ can be defined as $\{v_i \mid v \in P\}$.

There are three possibilities: only one colour occurs in P, precisely two colours occur in P or all three colours occur in P.
Note that the fact that Γ is a 2-IPP code implies that the forbidden configurations do no occur. (See theorem 3.2.)

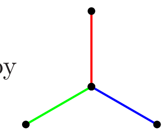
Case (i): all edges in P are blue.
This means all vertices in P have the same symbol on the first coordinate and all have different symbols on the second coordinate and on the third coordinate. Therefore $|P| = {q_2}' = {q_3}' \leq {q_1}' + {q_2}' + {q_3}' - 2$.

Case (ii): all edges in P are blue or green.
This means all vertices still have different third coordinates.
And so $|P| = {q_3}' \leq {q_1}' + {q_2}' + {q_3}' - 2$.

Case (iii): all three colours occur in P.
This is a more complicated situation.
First observe that P can only be of one type, namely three complete subgraphs of one of the colours each, with exactly one vertex in common.
An example:



This can symbolically be represented by



.

This is due to colour transitivity; if a and b are connected by a red edge and b and c are connected by a red edge, a and c must be too. Thus if you have a red subgraph Q of P then Q is complete. So if there is a green edge connected to one of the vertices of Q, you cannot connect a blue edge to any of the other vertices of Q, without creating a forbidden configuration.
The common vertex shall be called the *central vertex of* P.
Let s_1 be the size of the blue subgraph, s_2 the size of the green subgraph and s_3 the size of the red subgraph (all including the central vertex). Then the size of P is $s_1 + s_2 + s_3 - 2$.
Now, ${q_1}' = 1 + s_2 + s_3 - 2 = s_2 + s_3 - 1$, since for the blue subgraph only one symbol is needed, but for the green and the red subgraphs s_2 and s_3 symbols are needed but now the central vertex of P is counted three times.

Similarly, $q_2{}' = s_1+1+s_3-2 = s_1+s_3-1$ and $q_3{}' = s_1+s_2+1-2 = s_1+s_2-1$.
So $q_1{}' + q_2{}' + q_3{}' = 2s_1 + 2s_2 + 2s_3 - 3 = 2(s_1 + s_2 + s_3 - 2) + 1 = 2|P| + 1$.
Therefore $|P| = \frac{q_1{}'+q_2{}'+q_3{}'-1}{2} \leq q_1{}' + q_2{}' + q_3{}' - 2$ (since each of the $q_i{}' \geq 1$).

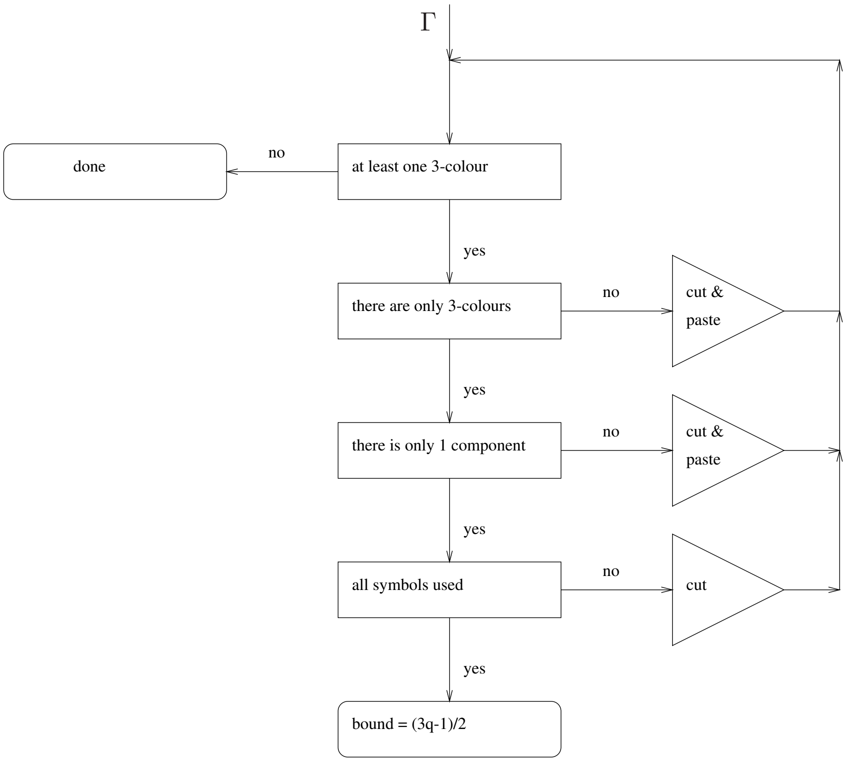
In each of the cases $|P| \leq q_1{}' + q_2{}' + q_3{}' - 2$.
Thus, since the alphabet is split disjointly over all the components, $|\Gamma| \leq q_1 + q_2 + q_3 - 2n$ where n is the number of components of the associated graph of Γ.
So in any case $|\Gamma| \leq q_1 + q_2 + q_3 - 2 \leq 3q - 2$.

Chapter 5

Transforming graphs

In this chapter I will show that it is possible to transform graphs, preserving certain properties.
The transformation will preserve: the size of the graph, the property that the graph is associated to a 2-IPP code (the forbidden configurations do not occur) and the (size of the) alphabet the code is defined over.
The goal is to transform a graph containing a connected component with all three colours into a graph without such a component.
As stated in section 4.2 one can regard different alphabets for all three coordinates and A can then be regarded as (A, A, A).
A connected component with all three colours will be denoted as a *3-colour*.
Remember that a 3-colour consists of three complete subgraphs, each of one of the three colours, incident at one vertex, the central vertex.

Let Γ be a c-IPP code of which the associated graph has at least one 3-colour.
Follow the scheme on the next page until Γ does not contain any 3-colours anymore.



Cut and paste works as follows.
Let P be one of the 3-colours.
Select a complete subgraph S of $\Gamma \setminus P$ of one colour, say colour i.
Let T be the subgraph of P of colour i, excluding the central vertex of P.
Now comes the actual 'cutting and pasting'.
Change the entries on the ith coordinate of all the vertices of T into the ith coordinate of the elements of S. (S and T have different ith coordinates because otherwise they would be connected.)
T is now disconnected from P while T and S have merged into a new S.
So P is now no longer a 3-colour (and neither has S become one).
It is obvious that no new symbols are used and no forbidden configurations were created either since T was not connected to any other vertex except the central vertex of P.
So a new graph is created of the same size as Γ that still is associated to a 2-IPP code Γ' which is still defined over the same alphabet.

Cut works a follows.
You are now in the situation that there is only the 3-colour P. But in one of the coordinates not all symbols are used, say in the ith coordinate.
Let T be the subgraph of P of colour i, excluding the central vertex of P.
Change the ith coordinate of all the vertices in T into an unused symbol.
This way T is disconnected from P.
So P is now no longer a 3-colour.
And just as with *cut and paste* the size of the graph did not change, no forbidden configurations were created and we are still working over the same alphabet.

As can be seen from the scheme, the situation that the graph only consist of a single 3-colour and all symbols are used cannot be converted to a graph without 3-colours.
From the proof of theorem 4.2 you can see that the size of a graph with only a 3-colour is $\frac{q_1+q_2+q_3-1}{2}$ which is at most $\frac{3q-1}{2}$. In fact, in this case it is exactly $\frac{3q-1}{2}$, since there were no symbols unused so $q_1 = q_2 = q_3 = q$.
Since $\frac{3q-1}{2} \leq 3q - 6\sqrt{q+1} + 6$ for $q \geq 7$ and I am interested in proving an upper bound of $3q - 6\sqrt{q+1} + 6$ for the maximal size of a 2-IPP code, this situation is taken care of.

So from now on I will assume graphs of 2-IPP codes not to contain 3-colours, since I will no longer have to consider the situation that cannot be converted.

Chapter 6

Proof of the optimal configuration of the trisection of a graph

In this chapter I will consider parts, instead of components as I did in section 4.2, of the associated graph of a 2-IPP code.
Remember that a part does not have to be connected and that it may contain components of size 1.
Let Γ be a 2-IPP code. Partition the associated graph of Γ into three parts, which may be empty.
I will show that the largest codes, given q, have associated graphs as follows: the first part contains blue and green edges only, the second part contains blue and red edges only and the third part contains green and red edges only.
I will do this by a simple but time consuming method: checking all possibilities. If you are willing to take my word for it you can easily skip this chapter although I think it is very instructive to look at least at the first few cases so you can understand better the way of looking at these graphs.
In the table on the next page I will enumerate all possibilities.
Some situations will be regarded the same. For instance, the situation that the first part consists of only blue edges, the second part consists of blue and green edges and the third part is empty will be regarded the same as the situation that the first part consists of only green edges, the second part consists of blue and green edges and the third part is empty.
However, the situation that the first part consists of only red edges, the sec-

ond part consists of blue and green edges and the third part is empty will be regarded a different situation.
The reason is lemma 3.3; coordinates can be permuted.
The situation that the first part consists of only blue edges, the second part consists of blue and green edges and the third part is empty will be denoted by $(blue, blue + green, \emptyset)$, a part with n isolated vertices will be denoted by n. So in this notation $(blue, blue+green, \emptyset)$ will be regarded the same as $(green, blue+green, \emptyset)$, but not the same as $(red, blue + green, \emptyset)$.
It follows from chapter 5 that I can assume that components with all three colours do not occur.
Note that isolated vertices can be placed in any of the three parts. This way isolated vertices can be put in a part that contains edges (of any colour) and so a part either contains edges or is empty. An exception is case I.

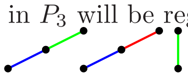
I	$(n, \emptyset, \emptyset)$
II	$(blue, \emptyset, \emptyset)$
III	$(blue, blue, \emptyset)$
IV	$(blue, blue, blue)$
V	$(blue, blue, green)$
VI	$(blue, green, \emptyset)$
VII	$(blue, green, red)$
$VIII$	$(blue + green, \emptyset, \emptyset)$
IX	$(blue + green, blue, \emptyset)$
X	$(blue + green, blue, blue)$
XI	$(blue + green, blue, green)$
XII	$(blue + green, blue, red)$
$XIII$	$(blue + green, red, \emptyset)$
XIV	$(blue + green, red, red)$
XV	$(blue + green, blue + green, \emptyset)$
XVI	$(blue + green, blue + green, blue)$
$XVII$	$(blue + green, blue + green, red)$
$XVIII$	$(blue + green, blue + green, blue + green)$
XIX	$(blue + green, blue + green, blue + red)$
XX	$(blue + green, blue + red, \emptyset)$
XXI	$(blue + green, blue + red, blue)$
$XXII$	$(blue + green, blue + red, green)$
$XXIII$	$(blue + green, blue + red, green + red)$

A little explanation on how the results below were achieved may be in place.

P_1, P_2 and P_3 will be the names of the three parts of (the graph of) Γ.
I will use the estimates from section 4.2 for the size of each part, but I will also use some information I have about the other two parts.

As an example I will discuss case XXII.

$(blue + green, blue + red, green)$
So P_1 contains edges only of colour blue and green, P_2 contains edges only of colours blue and red and P_3 contains edges only of colour green.
Note that isolated vertices can be placed in any part. For example those placed in P_3 will be regarded as green components of size 1. In a symbolic picture:



The sizes of P_1, P_2 and P_3 are found as in section 4.2:
$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = q_1{}''' = q_3{}'''$

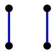
$q_3{}' + q_3{}''' \leq q - 2$ because, as you can see in the picture, P_2 must use at least two different symbols on the third coordinate; one for the vertex on the blue end and and one for the red part.
$q_2{}'' \leq q - 3$ because P_1 uses at least two different symbols on the second coordinate, just as P_2 does on the third coordinate, and P_3 uses at least one symbol on the second coordinate for the red component.
So:
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_2{}'' + q_3{}''' = q_3{}' + q_3{}''' + q_2{}'' \leq q - 2 + q - 3 = 2q - 5$

All estimates in the following cases are found in similar ways.

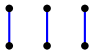
I
$|P_1| = n$
$|P_2| = 0$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = n \leq q$

II

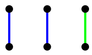
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = 0$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_2}' \leq q$

III 

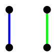
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = {q_2}'' = {q_3}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_2}' + {q_2}'' \leq q$

IV 

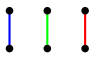
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = {q_2}'' = {q_3}''$
$|P_3| = {q_2}''' = {q_3}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_2}' + {q_2}'' + {q_2}''' \leq q$

V 

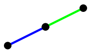
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = {q_2}'' = {q_3}''$
$|P_3| = {q_1}''' = {q_3}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_3}''' \leq q$

VI 

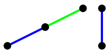
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = {q_1}'' = {q_3}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' \leq q$

VII 

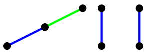
$|P_1| = {q_2}' = {q_3}'$
$|P_2| = {q_1}'' = {q_3}''$
$|P_3| = {q_1}''' = {q_2}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_1}''' \leq q - 1 + q - 3 = 2q - 4$

VIII 

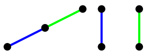
$|P_1| = q_3{}'$
$|P_2| = 0$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' \leq q$

IX 

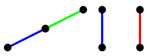
$|P_1| = q_3{}'$
$|P_2| = q_2{}'' = q_3{}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_3{}'' \leq q$

X 

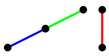
$|P_1| = q_3{}'$
$|P_2| = q_2{}'' = q_3{}''$
$|P_3| = q_2{}''' = q_3{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_3{}'' + q_3{}''' \leq q$

XI 

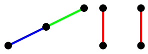
$|P_1| = q_3{}'$
$|P_2| = q_2{}'' = q_3{}''$
$|P_3| = q_1{}''' = q_3{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_3{}'' + q_3{}''' \leq q$

XII 

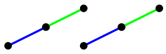
$|P_1| = q_3{}'$
$|P_2| = q_2{}'' = q_3{}''$
$|P_3| = q_1{}''' = q_2{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_3{}'' + q_1{}''' \leq q - 1 + q - 3 = 2q - 4$

XIII 

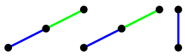
$|P_1| = q_3{}'$
$|P_2| = q_1{}'' = q_2{}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_1{}'' \leq q - 1 + q - 2 = 2q - 3$

XIV 

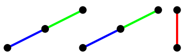
$|P_1| = {q_3}'$
$|P_2| = {q_1}'' = {q_2}''$
$|P_3| = {q_1}''' = {q_2}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_1}'' + {q_1}''' \leq q - 2 + q - 2 = 2q - 4$

XV 

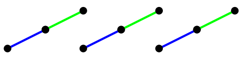
$|P_1| = {q_3}'$
$|P_2| = {q_3}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' \leq q$

XVI 

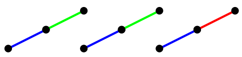
$|P_1| = {q_3}'$
$|P_2| = {q_3}''$
$|P_3| = {q_2}''' = {q_3}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_3}''' \leq q$

XVII 

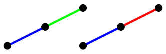
$|P_1| = {q_3}'$
$|P_2| = {q_3}''$
$|P_3| = {q_1}''' = {q_2}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_1}''' \leq q - 1 + q - 4 = 2q - 5$

XVIII 

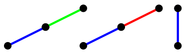
$|P_1| = {q_3}'$
$|P_2| = {q_3}''$
$|P_3| = {q_3}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_3}''' \leq q$

XIX 

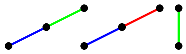
$|P_1| = {q_3}'$
$|P_2| = {q_3}''$
$|P_3| = {q_2}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = {q_3}' + {q_3}'' + {q_2}''' \leq q - 2 + q - 4 = 2q - 6$

XX 

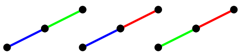
$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = 0$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_2{}'' \leq q - 2 + q - 2 = 2q - 4$

XXI 

$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = q_2{}''' = q_3{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_2{}'' + q_2{}''' \leq q - 4 + q - 2 = 2q - 6$

XXII 

$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = q_1{}''' = q_3{}'''$
$|\Gamma| = |P_1|+|P_2|+|P_3| = q_3{}'+q_2{}''+q_3{}''' = q_3{}'+q_3{}'''+q_2{}'' \leq q-2+q-3 = 2q-5$

XXIII 

$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = q_1{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_2{}'' + q_1{}''' \leq q - 4 + q - 4 + q - 4 = 3q - 12$

It follows from this (exhaustive) analysis that case XXIII offers (asymptotically) the best prospects for obtaining codes of large size.
In the next chapter I will derive a more precise upper bound on the size of 2-IPP codes with graphs of this type.

Chapter 7

The new upper bound

In this chapter I will give an upper bound on the size of a 2-IPP code that is almost best possible.
It will be $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$, exactly the size of the code constructed in section 4.1 when $q = r^2 + 2r$. Some technical details of the proof will be omitted here.
For this I will not regard situation XXIII from chapter 6, but a slightly more general case: $(\neg blue, \neg green, \neg red)$ which means that I will devide the graph of the code in three (possibly empty) parts and the first part will not contain blue edges, the second part will not contain green edges and the third part will not contain red edges.
So $|P_1| = q_1{}'$, $|P_2| = q_2{}''$ and $|P_3| = q_3{}'''$. (Notice that this is slightly different from chapter 6, but it is nothing more than a permutation of the three parts.)
Note that the algorithm in chapter 5 makes it possible to transform a graph that cannot be partitioned in $(\neg blue, \neg green, \neg red)$ into a graph that can be, preserving the properties as mentioned in chapter 5, except one case where we already have an upper bound $\frac{3q-1}{2}$.

7.1 Suitable schemes

Let Γ be the graph of a 2-IPP code.
Partition it in P_1, P_2 and P_3 as follows: $(\neg blue, \neg green, \neg red)$.
They can be arranged in a scheme as follows:

$$\begin{array}{ccc} P_1 & P_2 & P_3 \\ {q_1}' & {q_1}'' & {q_1}''' \\ {q_2}' & {q_2}'' & {q_2}''' \\ {q_3}' & {q_3}'' & {q_3}''' \end{array}$$ with all entries natural numbers ≥ 1.

There are a few inequalities that hold:
(a) ${q_1}' + {q_1}'' + {q_1}''' \leq q$
(b) ${q_2}' + {q_2}'' + {q_2}''' \leq q$
(c) ${q_3}' + {q_3}'' + {q_3}''' \leq q$
(d) ${q_2}' \cdot {q_3}' \geq {q_1}'$
(e) ${q_1}'' \cdot {q_3}'' \geq {q_2}''$
(f) ${q_1}''' \cdot {q_2}''' \geq {q_3}'''$
((d), (e) and (f) will be explained below.)
The goal is to maximize $|P_1| + |P_2| + |P_3| = {q_1}' + {q_2}'' + {q_3}'''$.

A scheme where these inequalities hold will be called a *suitable scheme*. The *size of a suitable scheme* will be defined as the sum of the diagonal elements.
Notice that the size of a suitable scheme is the same as the size of the 2-IPP code it belongs to.
At the end of this section I will show how a 2-IPP code can be constructed for a suitable scheme.

But first the explanation for (d), (e) and (f):
Consider P_1.
The size of P_1 is ${q_1}'$.
Now disregard all green edges of P_1 for a moment.
There will be ${q_3}'$ red components (inluding those of size 1).
Let C^3 be the enumerated set of all these red components.
Instead, now disregard all red edges of P_1 for a moment.
Similarly there will be ${q_2}'$ green components.
Let C^2 be the enumerated set of all those green components.
Obviously, $|P_1| = \sum_{i=1}^{{q_3}'} |{C^3}_i| \leq {q_3}' \cdot \max_i(|{C^3}_i|)$.
$\max_i(|{C^3}_i|) \leq |C^2|$ since if $\max_i(|{C^3}_i|) > |C^2|$ at least one red edge would connect two vertices of the same green component and that would give the forbidden configuration of double edges (see theorem 3.2).
So ${q_1}' \leq {q_3}' \cdot \max_i(|{C^3}_i|) \leq {q_3}' \cdot {q_2}'$.
Similarly ${q_2}' \leq {q_1}'' \cdot {q_3}''$ and ${q_3}' \leq {q_1}''' \cdot {q_2}'''$.

Suppose $|P_1| + |P_2| + |P_3|$ is maximal for a suitable scheme.
Then we may assume that (a), (b) and (c) are in fact equalities:

At least one of (a), (b) or (c) has to be an equality, otherwise I am actually working over a smaller alphabet.

Suppose (a) is an equality and (b) and (c) are not.
There is at least one symbol that is used more than once on the first coordinate since, as I mentioned below lemma 3.1, I assume codes to consist of at least $q + 1$ codewords.
So there is at least one blue edge and since we are considering 2-IPP codes, such a blue edge is not part of a forbidden configuration.
Then codewords can be added; as a first coordinate take any symbol that is used at least twice on the first coordinate and unused symbols in the second and third coordinate. This way no forbidden configurations are created. (An already exsisting blue (sub)component is enlarged a little and no edges of other colours were added.)
This is a contradiction since I already assumed the code to be of maximal size.
So there must be at least two equalities.

Suppose (c) is not an equality and the other two are.
As mentioned above, the size of the code is at least $q+1$ so on every coordinate there is always at least one symbol used twice.
Now, take a red edge, say between vertices v_1 and v_2.
Change the third coordinate of v_1 into a unused symbol. This way the red edge between v_1 and v_2 is broken and all other red edges connected to v_1 too.
This way no forbidden configuration can emerge, so the code remains a 2-IPP code and the size does not change either.
This can be repeated until all symbols are used in the third coordinate too.

So (a), (b) and (c) can be assumed to be equalities.

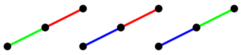
7.1.1 Creating a 2-IPP code from a suitable scheme

Below I will only give the construction of a part of a 2-IPP code for a specific suitable scheme, the one belonging to $q = 18$ in the table at the end of this chapter, but it will be clear how this construction can be extended to the other two parts of the code and to suitable schemes in general.

First I will show how to make a graph from a suitable scheme.

For $q = 18$ I gave the scheme $\begin{array}{lll} 11 & 3 & 4 \\ 4 & 11 & 3 \\ 3 & 4 & 11 \end{array}$.

I will use this to create a graph of the type



with

$$\begin{array}{lll} q_1{}' & q_1{}'' & q_1{}''' \\ q_2{}' & q_2{}'' & q_2{}''' \\ q_3{}' & q_3{}'' & q_3{}''' \end{array} = \begin{array}{lll} 11 & 3 & 4 \\ 4 & 11 & 3 \\ 3 & 4 & 11 \end{array} .$$

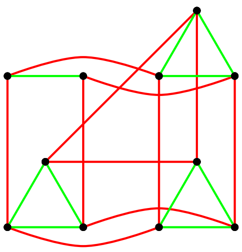
The idea comes from the proof of inequalities (d), (e) and (f).

Part P_1 must consist of 11 vertices, using only 4 symbols in the second coordinate and 3 symbols in the third coordinate.

Make four roughly equally large green components, so in this case three of size 3 and one of size 2.

Now make three red ‘components’ by connecting ‘the first’ vertices of the green components with red edges, the ‘second’ vertices of the green components and so on.

It will look like this:



This picture is not completely accurate; each of the red 4-cycles should be a complete graph. For clarity reasons they were deliberately drawn as 4-cycles instead of complete graphs.

It will now be clear how the other two parts of the graph are constructed and generally how any of the schemes in the third column is turned into the graphs of a 2-IPP code.

Next I will show how a 2-IPP code can be constructed from an (appropriate) graph:

Choose one vertex to be $(1, 1, 1)$. Any vertex connected to this vertex by a blue edge also has 1 on its first coordinate and so on for the other colours.

Now some vertices have (some of) their cooridnates filled in and the rest is still blank.
Continue choosing unused symbols for blank coordinates and use the edges to fill in the coordinates of connected vertices.
Keep doing this until all vertices have all their coordinates filled in.

Notice that the size of a code constructed from a suitable scheme is the same as the size of that suitable scheme.

7.2 Semi-suitable schemes

It proves very difficult to optimize the size of suitable schemes in order to find the optimal size for 2-IPP codes. So a new type of schemes is introduced: a *semi-suitable scheme*.
These are schemes where the same equalities hold as for the suitable schemes, but with real numbers, ≥ 1, as entries instead of natural numbers.
Again, the size is defined as the sum of the diagonal elements.

We already know that equations (a), (b) and (c) are in fact equalities for (semi-)suitable schemes of maximal size.
For semi-suitable schemes of maximal size also (d), (e) and (f) are equalities:
Let the scheme be of a code of maximal size.
Suppose ${q_2}' \cdot {q_3}' > {q_1}'$.
Without loss of generality I may assume that ${q_2}', {q_3}' > 1$.
Choose $\epsilon_1, \epsilon_2 > 0$ so that $({q_2}' - \epsilon_1)({q_3}' - \epsilon_2) = {q_1}'$.
Let $\epsilon_3 = \min(\epsilon_2, \epsilon_1 \cdot {q_1}''')$.
Then the scheme below is also a suitable scheme.

P_1	P_2	P_3
${q_1}'$	${q_1}''$	${q_1}'''$
${q_2}' - \epsilon_1$	${q_2}''$	${q_2}''' + \epsilon_1$
${q_3}' - \epsilon_2$	${q_3}'' + \epsilon_2 - \epsilon_3$	${q_3}''' + \epsilon_3$

But now the size of the code has increased by ϵ_3 which is a contradiction with the original code being of maximal size.
So (d), (e) and (f) must be equalities for a code of maximal size (if the numbers do not have to be natural numbers).

Since I am interested in finding the maximal size, I can assume all inequalities to be equalities.

In the next section I will derive a semi-suitable schemes of maximal size

and I will show how to make a suitable scheme from this semi-suitable scheme, without decreasing the size much.
So, using subsection 7.1.1, I will be able to create a 2-IPP code of almost maximal size.

7.3 The semi-suitable scheme of maximal size

Theorem 7.1 *The semi-suitable scheme of maximal size is the symmetric case:*

$$\begin{array}{lll} q-2\sqrt{q+1}+2 & \sqrt{q+1}-1 & \sqrt{q+1}-1 \\ \sqrt{q+1}-1 & q-2\sqrt{q+1}+2 & \sqrt{q+1}-1 \\ \sqrt{q+1}-1 & \sqrt{q+1}-1 & q-2\sqrt{q+1}+2 \end{array}$$

Proof:
Suppose that

$$\begin{array}{lll} x_1 & z_2 & . \\ . & x_2 & . \\ . & . & x_3 \end{array}$$

is a semi-suitable scheme for which the size $x_1 + x_2 + x_3 = C$.
The scheme stated in the theorem shows that a semi-suitable scheme with size $C \geq 3(q - 2\sqrt{q+1} + 2) = C_0$ exists.

The conditions (a)—(f) lead to a quadratic equation for z_2.
To prove that there are no semi-suitable schemes with size $C > C_0$, write down the equations (of degree 4) in x_1, x_2, x_3 that guarantee that the solutions to the quadratic equation for z_2 are real and the equations that imply that the scheme is semi-suitable, in particular that all entries are ≥ 1.
It can be shown that there are no solutions to these equations with $x_1+x_2+x_3 > C_0$.
For details see *A sharp bound for* 2*-IPP codes of length* 3 by Wieb Bosma and myself.
QED
This solution generally consists of real numbers. Only for $q = r^2+2r$ $(r \in \mathbb{N}_{>0})$ we will have natural numbers and for those cases we also have a construction. (See section 4.1.)

Corollary 7.2 *We now have an upper bound* $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$ *(the floor of the size of the semi-suitable scheme from the theorem) for the size of* 2*-IPP codes.*

Proof:
The set of suitable schemes is a subset of the set of semi-suitable schemes. So $3q - 6\sqrt{q+1} + 6$ is also an upper bound for suitable schemes. But suitable schemes obviously can only have natural numbers as a size so $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$ is an upper bound for their size. In subsection 7.1.1 I showed it is possible to create a 2-IPP code from a suitable scheme of the same size. So $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$ is an upper bound for the size of 2-IPP codes.
QED

7.3.1 Making a suitable scheme from the semi-suitable scheme of maximal size

In this subsection I will construct a suitable scheme from the maximal semi-suitable scheme and the size will decrease less than 3, so this will give codes of size at most 2 less than the upper bound.

As can be seen from the maximal solution for real numbers, it is of the form:

$$\begin{pmatrix} e & d & d \\ d & e & d \\ d & d & e \end{pmatrix} \quad \text{with} \quad \begin{array}{l} e, d > 0 \\ e + 2d = q \\ d^2 = e \end{array}$$

Assume $e, d \notin \mathbb{N}$.
Notice that from $e + 2d \in \mathbb{N}$ and $d^2 = e$ it follows that either both are natural numbers or both are not.
Let $E = \lfloor e \rfloor$, $\epsilon = \{e\}$ ($\{x\} = x - \lfloor x \rfloor$, the fractional part), $D = \lfloor d \rfloor$ and $\delta = \{d\}$.
So $e = E + \epsilon$ and $d = D + \delta$.
Note that $0 < \epsilon, \delta < 1$.
$q = e + 2d = E + 2D + \epsilon + 2\delta$ so $\epsilon + 2\delta \in \mathbb{N}$.
Since $0 < \epsilon, \delta < 1$ we have $0 < \epsilon + 2\delta < 3$. So $\epsilon + 2\delta = 1$ or 2.
I want to make a scheme of natural numbers out of the scheme of real numbers which still satisfies the inequalities (a)—(f). A logical choice is rounding (up and/or down).

I distinguish three cases:

Suppose $\delta > \frac{1}{2}$:
Then $\epsilon + 2\delta > \epsilon + 1 > 1$, so $\epsilon + 2\delta = 2$.
So $q = e + 2d = E + 2D + \epsilon + 2\delta = E + 2D + 2 = E + 2(D + 1)$.
And $(D + 1)^2 > (D + \delta)^2 = E + \epsilon > E$.

So
$$\begin{pmatrix} E & D+1 & D+1 \\ D+1 & E & D+1 \\ D+1 & D+1 & E \end{pmatrix}$$
is a suitable scheme.

Suppose $\delta = \frac{1}{2}$:
Then $E + \epsilon = (D+\delta)^2 = (D+\frac{1}{2})^2 = D^2 + D + \frac{1}{4}$ and since $D^2 + D \in \mathbb{N}$ would $\epsilon = \frac{1}{4}$.
But then $\epsilon + 2\delta = 1\frac{1}{4} \notin \mathbb{N}$.
So $\delta = \frac{1}{2}$ is not an option. (As a matter of fact, δ and ϵ both cannot be rational.)

Suppose $\delta < \frac{1}{2}$:
Then $\epsilon + 2\delta < \epsilon + 1 < 2$, so $\epsilon + 2\delta = 1$.
So $q = e + 2d = E + 2D + \epsilon + 2\delta = E + 2D + 1 = E + D + (D+1)$.
$D \cdot (D+1) = D^2 + D = (D+\frac{1}{2})^2 - \frac{1}{4} > (D+\delta)^2 - \frac{1}{4} = E + \epsilon - \frac{1}{4} > E - \frac{1}{4}$.
So $D \cdot (D+1) \geq E$
So
$$\begin{pmatrix} E & D & D+1 \\ D+1 & E & D \\ D & D+1 & E \end{pmatrix}$$
is a suitable scheme.

As stated before it is quite easy to construct a 2-IPP code for such suitable schemes.
Furthermore, the size of such a code would be $3E = 3\lfloor e \rfloor$ and the upper bound is $\lfloor 3e \rfloor$.
$\lfloor 3e \rfloor = \lfloor 3E + 3\epsilon \rfloor \geq \lfloor 3E + 0 \rfloor = 3E$
$\lfloor 3e \rfloor = \lfloor 3E + 3\epsilon \rfloor < \lfloor 3E + 3 \rfloor = 3E + 3 \Leftrightarrow 3E \geq \lfloor 3e \rfloor - 2$

This leads to the following theorem.

Theorem 7.3 *Given q, it is possible to make 2-IPP codes with size at most 2 less than the upper bound $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$.*

7.4 Examples of suitable schemes of maximal size

In the table below I give suitable schemes of maximal size for several values of q.
Generally, they are quite symmetric over the skew diagonal.
I know these suitable schemes are of maximal size because I used a computer program that calculated all possibilities.
I also give the size of the semi-suitable schemes of maximal size, from theorem 7.1, for those q.
Most sizes of the suitable schemes reach the upper bound, but as can be expected not all. However, their size is at most 1 below the upper bound.
It is my conjecture that there is a general construction possible that achieves always 2-IPP codes with size at most 1 below the upper bound and mostly 2-IPP codes with size exactly the upper bound.

r	q	$\begin{matrix} x & y'' & z' \\ x' & y & z'' \\ x'' & y' & z \end{matrix}$	$\lvert\Gamma\rvert$	$3q - 6\sqrt{q+1} + 6$
1	3	$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$	3	3
	4	$\begin{matrix} 2 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \end{matrix}$	4	4.584
	5	$\begin{matrix} 2 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \end{matrix}$	6	6.303
	6	$\begin{matrix} 4 & 1 & 1 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{matrix}$	8	8.125
	7	$\begin{matrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{matrix}$	9	10.029
2	8	$\begin{matrix} 4 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 4 \end{matrix}$	12	12

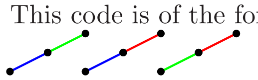
r	q	x y'' z' / x' y z'' / x'' y' z	$\|\Gamma\|$	$3q-6\sqrt{q+1}+6$
	9	5 2 2 3 4 2 2 3 4	13	14.026
	10	6 2 2 2 6 2 3 3 4	16	16.100
	11	6 2 3 3 6 2 2 3 6	18	18.215
	12	8 2 2 3 6 3 3 3 6	20	20.367
	13	8 3 2 2 8 3 4 3 6	22	22.550
	14	8 3 3 3 8 3 3 3 8	24	24.762
3	15	9 3 3 3 9 3 3 3 9	27	27
	16	10 3 3 4 9 3 3 4 9	28	29.261
	17	11 3 1 3 11 3 4 4 9	31	31.544
	18	11 3 4 4 11 3 3 4 11	33	33.847
	19	12 3 4 4 12 3 3 4 12	36	36.167
	20	14 3 3 4 12 4 4 4 12	38	38.505

r	q	x x' x''	y'' y y'	z' z'' z	$\lvert\Gamma\rvert$	$3q-6\sqrt{q+1}+6$
		14	4	3		
	21	3	14	4	40	40.858
		5	4	12		
		14	4	4		
	22	4	14	4	42	43.225
		4	4	14		
		15	4	4		
	23	4	15	4	45	45.606
		4	4	15		
		16	4	4		
4	24	4	16	4	48	48
		4	4	16		
		17	4	4		
	25	5	16	4	49	50.406
		4	5	16		
		18	4	4		
	26	4	18	4	52	52.823
		5	5	16		
		18	4	5		
	27	5	18	4	54	55.251
		4	5	18		
		19	4	5		
	28	5	19	4	57	57.689
		4	5	19		
		20	4	5		
	29	5	20	4	60	60.137
		4	5	20		
		22	4	4		
	30	5	20	5	62	62.593
		5	5	20		
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$

Chapter 8

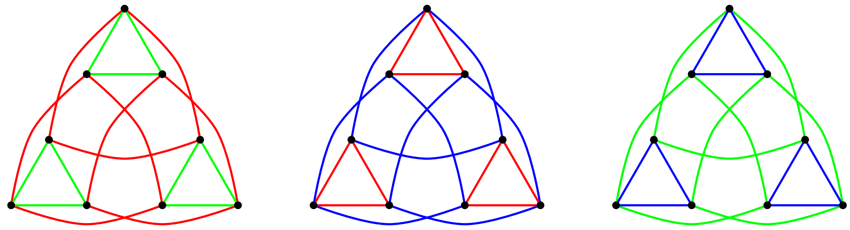
Conclusion

In this chapter I will recall the results of prior chapters and combine them to the main result of this thesis.
In section 4.1 I constructed a code of size $3q - 6\sqrt{q+1} + 6$ for $q = r^2 + 2r$ $(r \in \mathbb{N}_{>0})$.
This code is of the form:



$|P_1| = q_3{}'$
$|P_2| = q_2{}''$
$|P_3| = q_1{}'''$
$|\Gamma| = |P_1| + |P_2| + |P_3| = q_3{}' + q_2{}'' + q_1{}''' \leq q - 4 + q - 4 + q - 4 = 3q - 12$
In chapter 6 I have shown that codes of that form are of maximal size.
In chapter 7 I derive the upper bound $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$ for such codes, but I needed chapter 5 to remove some obstacles.
Essential, also, is theorem 3.2, which gives the connection between 2-IPP codes and their associated graphs.
So, it appears that the code constructed in section 4.1 is actually of maximal size.

An example of the graph of such a code of maximal size (on the next page):



$q = 15$
$|\Gamma| = 27$

I also suggest a construction for codes when $q \neq r^2 = 2r$ which I expect to be at most 1 below $\lfloor 3q - 6\sqrt{q+1} + 6 \rfloor$, but I cannot prove that yet.
So generally I expect that codes exist of size approximately $3q - 6\sqrt{q+1} + 6$.

Chapter 9

Ideas for further research

- Study the pattern in the table in chapter 7 and derive an exact formula for the maximal size of 2-IPP codes of length 3 for every q.
- Investigate whether graphs associated to 2-IPP codes of length 4 are of use too.

References

Henk D. Hollmann, Jack H. van Lint, Jean-Paul Linnartz and Ludo M.G.M. van Tolhuizen, *On codes with the identifiable parent property*, J. Combin. Theory**82** (1998), 121–133.

Mink G. Assies, Wieb Bosma, *A sharp bound for* 2*-IPP codes of length* 3, to appear.