THE UNIVERSITY OF WARWICK

Original citation:
Beynon, Meurig (1987) Monotone Boolean functions as combinatorially piecewise linear maps. University of Warwick. Department of Computer Science. (Department of Computer Science Research Report). (Unpublished) CS-RR-109

Permanent WRAP url:
http://wrap.warwick.ac.uk/60805

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

A note on versions:
The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.For more information, please contact the WRAP Team at: publications@warwick.ac.uk

warwickpublicationswrap

highlight your research

http://wrap.warwick.ac.uk/

Research Report 109

MONOTONE BOOLEAN FUNCTIONS AS COMBINATORIALLY PIECEWISE LINEAR MAPS

Meurig Beynon

(RR109)

The class of combinatorially piecewise linear (cpl) maps was first introduced to solve a geometric problem concerning the representability of piecewise linear functions as pointwise maxima of minima of linear functions. Such maps correspond in a canonical fashion to monotone boolean functions. This paper describes how a monotone boolean function in n variables whose prime implicants and prime clauses are non-trivial defines a partition of the symmetric group on n symbols into a set of "singular cycles" representing relations between transpositions of adjacent symbols. Several possible approaches to the classification of such cycles are described, and some characteristic properties of singular cycles are identified. The potential computational significance of singular cycles is indicated with reference to new combinatorial models for monotone boolean formulae and circuits that arise directly form the appropriate theory of computational equivalence and replaceability. The prospects for application to monotone boolean complexity are briefly examined. A catalogue of known relations is included as an Appendix.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

November 1987

Monotone boolean functions as combinatorially piecewise linear maps

Introduction

This paper is concerned with an unusual model for monotone boolean functions. The original motivation for introducing this model [1] was the proof of a geometric theorem concerning the representability of piecewise linear functions as pointwise maxima of minima of linear functions. The main motivation for subsequent research has been a search for combinatorial structure of monotone boolean functions that has computational significance. The results so far obtained are for the most part quite elementary, but are indicative of a rich combinatorial context for examining monotone boolean functions in which the algebraic structure of the free distributive lattice FDL(n) and the combinatorial geometry of a standard presentation Γ_n of the symmetric group S_n are central ingredients. It will be shown in particular that monotone boolean functions are naturally associated with a special class of relations in Γ_n, and that there is a canonical partition of the symmetric group associated with almost every monotone boolean function. A full understanding of the nature and significance of these partitions has so far proved elusive, and the principal purpose of this paper is to organise concepts and minor results that have been derived in the search for a formal classification. For the present, in view of the lack of an adequate theoretical understanding, little work has been done on potential applications. The research into computational equivalence for monotone boolean functions reported in [2] originated with the study of cpl maps, and some connections will be described (cf §8 Proposition 9 below). Some of the longer term objectives of this research are indicated in the final section of the paper; applications to monotone boolean function complexity, computational geometry and possibly concurrent program design may be anticipated.

The paper is divided into 10 sections. §1 defines the class of combinatorially piecewise linear (cpl) maps $S_n \rightarrow \{1,2,...,n\}$, and outlines its close connection with FDL(n), the lattice of monotone boolean functions in n variables. Throughout the paper, a standard presentation Γ_n of the symmetric group S_n is interpreted geometrically as a Cayley diagram, and the concept of singular edges in Γ_n associated with a cpl map is introduced. §2 outlines a duality for cpl maps that relates "value" and "rank". §3 explores the geometry of singular edges, establishing the presence of singular chains and cycles that have a special geometric form, and that in general define a partition of Γ_n. §4 indicates how the disposition of singular edges can be described with reference to the disjunctive and conjunctive normal forms of a monotone boolean function. §5 considers the structure of singular cycles from a group theoretic perspective, and introduces a useful notation that assists informal classification. §6 examines the prospects for generating singular cycles as group relations, and hence deriving monotone boolean functions with interesting computational characteristics. This motivates the consideration in §7 of a class of monotone boolean functions of a particularly simple kind defined by alternating sums and products. §8 describes a new perspective on boolean formulae and circuits that arises directly from the theory of computational equivalence and replaceability developed in [2], and indicates how the presence of particular singular cycles potentially has computational significance. In particular, a correspondence between singular cycles and configurations of prime implicants and prime clauses is described. §9 develops this theme further, and describes how the geometry of singular cycles is affected by basic circuit building operations. §10 identifies some key unresolved issues, and includes some speculation on future developments and applications.

In the interests of brevity and readability, many proofs are left to the reader: most of the propositions are simply intended to summarise observations that require little proof. To assist the reader, a large number of illustrative diagrams and examples is included. In one sense, the problem of classification of singular cycles is a central subject of the paper, and a catalogue of known cycles is included in an Appendix.

§1 Combinatorially piecewise-linear (cpl) maps

Let S_n denote the symmetric group of permutations of $\{1,2,...,n\}$. S_n has a standard presentation relative to the generating set consisting of the n-1 transpositions $\tau_1, \tau_2, ..., \tau_{n-1}$, where τ_i interchanges i and i+1. The explicit set of relations which defines this presentation is:

$$\tau_i^2=1\ (1\le i\le n-1);\ (\tau_i\tau_{i+1})^3=1\ (1\le i\le n-2);\ (\tau_i\tau_j)^2=1\ (2\le i+1<j\le n-1).$$

The Cayley diagram associated with this presentation is then an (n-1)-edge-coloured combinatorial graph Γ_n in which the vertices are in 1-1 correspondence with permutations in S_n and there is a bidirected edge of colour i from ρ to σ if and only if $\sigma=\tau_i\rho$ (using the convention whereby "$\tau_i\rho$" denotes "τ_i followed by ρ"). Figure 1 depicts the Cayley diagram Γ_4. Departing from the conventions defined in [1] (to avoid overloading the term "vertex"), a subset S of $\{1,2,...,n\}$ is said to be a **prefix** of the permutation σ in S_n if $S = \{1.\sigma, 2.\sigma,..., |S|.\sigma\}$.

A map $S_n\to\{1,2,...,n\}$ (of degree n) is **combinatorially piecewise-linear (cpl)** if it corresponds to an n-vertex-colouring of Γ_n in which vertices ρ and σ that are adjacent via an i coloured edge either have the same colour or have colours selected from the set $\{i.\sigma, (i+1).\sigma\}= \{i.\rho, (i+1).\rho\}$. The class of maps was introduced in [1] in connection with the study of representation of piecewise-linear functions as pointwise maxima of minima of linear functions. There is a 1-1 correspondence between cpl maps of degree n and elements of FDL(n), the finite distributive lattice freely generated by n generators $x_1, x_2,..., x_n$. In one direction, this correspondence is defined by associating with the cpl map F the function from the lattice of subsets of $\{1,2,...,n\}$ to $\{0,1\}$ which maps S to 1 if and only if $F(\sigma)\in S$ whenever S is a prefix of σ (cf [1]). The inverse is then defined by mapping the monotone boolean function f in FDL(n) to the cpl map F such that $F(\sigma)= r.\sigma$, where

$$f(\{1.\sigma, 2.\sigma,..., (r-1).\sigma\})=0 \text{ and } f(\{1.\sigma, 2.\sigma,..., r.\sigma\})=1.$$

Figure 2 depicts the vertex colouring of the Cayley diagram Γ_4 which is associated with the monotone boolean function $x_1x_2+x_2x_3x_4+x_1x_4$ in FDL(4).

Given a cpl map F, an edge $(\sigma,\sigma'=\tau_r\sigma)$ is **singular** for F if $F(\sigma)$ – and necessarily also $F(\sigma')$ - is in the set $\{r.\sigma,(r+1).\sigma\}$. Three types of singular edge can be distinguished:

(1) a positive (+) singularity, when $F(\sigma)=r.\sigma$, $F(\sigma')= (r+1).\sigma$,
   or equivalently $F(\sigma')=r.\sigma'$, $F(\sigma)=(r+1).\sigma'$

(2) a negative (-) singularity, when $F(\sigma)=(r+1).\sigma$, $F(\sigma')=r.\sigma$,
   or equivalently $F(\sigma')= (r+1).\sigma'$, $F(\sigma)= r.\sigma'$

(3) a zero (0) singularity, when $F(\sigma')=F(\sigma)$.

If F is a cpl map such that $F(\sigma)= r.\sigma$, then r is the **rank** of F at σ. The positive and negative singular edges for F can then be characterised as those edges across which F changes in value but not rank, and the zero singular edges as those across which F changes in rank but not in value (Figure 3). The brief digression on dual cpl maps that follows gives another perspective on this concept of singularity.

§2 Dual cpl maps

There are two Cayley diagrams associated with the presentation of S_n defined above; Γ_n, in which adjacency of vertices is defined by multiplication by a transposition τ_i on the left, and a dual graph Γ_n^*, in which adjacency is defined via multiplication by a transposition τ_i on the right. (Note that different pairs of permutations are adjacent in Γ_n and Γ_n^*.) A map $S_n \to \{1,2,...,n\}$ (of degree n) is **dual combinatorially piecewise-linear (dual cpl)** if it corresponds to an n-vertex-colouring of Γ_n^* in which vertices ρ and σ that are adjacent via an i coloured edge either have the same colour or have colours selected from the set $\{i, i+1\}$. As is the case for cpl maps, there is a 1-1 correspondence between elements of FDL(n) and dual cpl maps. Given $f \in$ FDL(n), there is an associated dual cpl map F^*: $S_n \to \{1,2,...,n\}$ defined as follows:

Let T be the total ordering T of $\{1,2,...,n\}$ defined by $1>2>...>n$, and let $F^*(\sigma)$ be the element to which $f \in$ FDL(n) is mapped by the canonical lattice homomorphism Π_σ:FDL(n)$\to$T (induced by freeness) under which $i.\sigma$ is the image of the generator x_i of FDL(n). It follows that

Lemma 1: F^* is a dual cpl map. Moreover, if F is the cpl map associated with $f \in$ FDL(n), then

$$F^*(\sigma) = F(\sigma^{-1}).\ \sigma$$

Proof:

Suppose that $\sigma \in S_n$. The free generators $x_1, x_2,..., x_n$ can be viewed as inputs to a monotone boolean circuit computing f. Suppose that all the inputs are initially assigned 0, and that they are systematically assigned 1 ("switched on") in such a way that $x_{t.\sigma}$ becomes 1 at time t. $F(\sigma)$ is then the index $s.\sigma$, where the output f is first 1 at time s. To prove Lemma 1, it will suffice to prove that $F^*(\sigma)$ is the *time* at which the output f is switched on if the inputs are switched on in the order σ^{-1}.

Let π_i be the map $T \to \{0,1\}$ defined by $\pi_i(j)=1$ iff $j \leq i$. Consider the parametrised family of composite maps of the form

$$\pi_i\Pi_\sigma : \text{FDL}(n) \to \{0,1\}$$

where $\sigma \in S_n$, and i ranges from 1 to n. It is evident that $F^*(\sigma)$ is the least i such that $\pi_i\Pi_\sigma(f)$ is 1. On the other hand, $\pi_i\Pi_\sigma(f)$ can also be interpreted as

$$f(\alpha_1, \alpha_2,..., \alpha_n), \text{ where } \alpha_j \in \{0,1\}, \text{ and } \alpha_j=1 \text{ iff } j=r.\sigma \text{ and } r\leq i$$

ie as the time at which f is switched on when switching on the inputs $x_1, x_2,..., x_n$ in the order prescribed by σ^{-1}.

It remains to show that F^* is a dual cpl map. To this end, consider vertices ρ and σ of Γ_n^* that are adjacent via an i coloured edge. Note first that $F^*(\rho) \in \{i,i+1\}$ iff $F^*(\sigma) \in \{i,i+1\}$. To see this, observe that $F^*(\rho) = F(\tau_i\, \sigma^{-1})\, .\sigma\tau_i$, so that

$$F^*(\rho) \in \{i,i+1\} \text{ iff } F(\tau_i\, \sigma^{-1}) \in \{i.\sigma^{-1}, (i+1).\sigma^{-1}\} \text{ iff } F(\sigma^{-1}) \in \{i.\sigma^{-1}, (i+1).\sigma^{-1}\}$$

since F is a cpl map. Since $F^*(\sigma) = F(\sigma^{-1}).\sigma$, the latter condition is equivalent to $F^*(\sigma) \in \{i,i+1\}$. Similarly, if $F^*(\sigma) \notin \{i,i+1\}$, then $F(\sigma^{-1}) \notin \{i.\sigma^{-1}, (i+1).\sigma^{-1}\}$, so that $F(\tau_i\, \sigma^{-1})=F(\sigma^{-1})$, and

$$F^*(\rho) = F(\tau_i\, \sigma^{-1}).\sigma\tau_i = F(\sigma^{-1}).\sigma = F^*(\sigma).$$

Corollary 1.1: F and F^* also satisfy the dual relationship $F(\sigma) = F^*(\sigma^{-1}).\sigma$

Figure 4 depicts the dual cpl map associated with the monotone boolean function $x_1x_2+x_2x_3x_4+x_1x_4$ in FDL(4), which can be compared with Figure 2. In pictorial terms, Lemma 1 and Corollary 1.1 express the fact that if Figures 2 and 4 are superimposed, then at each vertex the value of the function specified in the one diagram is the rank of the function specified in the other. If the concept of a singular edge for the dual cpl map F^* in $\Gamma_n{}^*$ is defined (by analogy with that for cpl maps) by adjacent vertices at which either the rank or the value of F^* changes, the 1-1 correspondence between Γ_n and $\Gamma_n{}^*$ that matches σ and σ^{-1} respects adjacency, and defines a 1-1 correspondence between the singular edges of F and those of F^*. In this correspondence: zero singular edges for F correspond to edges at which the value F^* changes but the rank remains the same: positive and negative singular edges for F to edges at which the rank of F^* changes but the value remains the same. Moreover, a positive (respectively negative) singular edge for F corresponds to an edge $(\sigma,\tau\sigma)$ in $\Gamma_n{}^*$ at which the value of F^* is the greater (respectively smaller) of the indices in the transposition τ under the ordering T.

§3 Chains and cycles of singular edges

As Figure 2 illustrates, the singular edges of a cpl map are generally associated with a **cycle**, consisting of a simple circuit in Γ_n representing a relation in the symmetric group S_n. In fact, if F is the cpl map corresponding to an element $f \in$ FDL(n) which is non-comparable with an input generator, the singular edges for F are disposed in disjoint cycles which partition Γ_n. The "1-dimensional structure" defined by singular edges is the central subject of this paper; though this appears to offer scope for interesting and challenging analyses, and there are some indications of computational significance, it will become clear that this structure cannot in itself characterise the computational complexity of a monotone boolean function.

To describe singular cycles formally, it is convenient to consider subgroups of S_n, viz:

$$K_{rs} = <\tau_r, \tau_s> \text{ where } |r-s| > 1 \text{ and } 1\le r,s\le n-1, \text{ and } T_r = <\tau_{r-1}, \tau_r> \text{ where } 2\le r\le n-1.$$

Given a permutation σ of S_n, the cosets $K_{rs}\,\sigma$ and $T_r\,\sigma$ are cycles of edges in Γ_n, as depicted in Figure 5, in which $\tau=\tau_r$, $\tau'=\tau_{r-1}$ and $\tau''=\tau_s$. Note that $T_r\,\sigma$ is the set of permutations obtainable from σ by permuting

$$(r-1).\sigma,\ r.\sigma,\ (r+1).\sigma$$

whilst $K_{rs}\,\sigma$ is the set of permutations obtainable from σ by permuting

$$r.\sigma,\ (r+1).\sigma \text{ and } s.\sigma,\ (s+1).\sigma$$

independently. The geometrical relationships between cosets of this type is then as follows:

Lemma 2: Either two cycles are edge and vertex disjoint, or they correspond to cosets having a common element σ. In the latter case, the possible intersections up to symmetry are

$$T_r\sigma \cap T_s\sigma = \{\sigma\} \text{ if } s \ne r+1 \;;\; T_r\sigma \cap T_{r+1}\sigma = \{\sigma, \tau_r\sigma\}$$

$$T_r\sigma \cap K_{ij}\sigma = \{\sigma\} \text{ if } i \ne r \text{ and } j \ne r \;;\; T_r\sigma \cap K_{rj}\sigma = \{\sigma, \tau_r\sigma\}$$

$$K_{ij}\sigma \cap K_{rs}\sigma = \{\sigma\} \text{ if } \{i,j\}\cap\{r,s\}=\emptyset \;;\; K_{rj}\sigma \cap K_{rs}\sigma = \{\sigma, \tau_r\sigma\} \text{ if } j \ne s$$

Proof: Exercise to the reader.

Proposition 3:

Let f be a cpl map $\Gamma_n \to \{1,2,...,n\}$. Then

(i) f has two singular edges at those vertices σ for which $1<rk(f(\sigma))<n$, and one singular edge at those vertices σ for which $rk(f(\sigma))$ is 1 or n

(ii) f is either non-singular on a $K_{rs}\,\sigma$ or has singularities of the same type (positive, negative or zero) on precisely one pair of opposite edges

(iii) f is either non-singular on a $T_r\,\sigma$ or has singularities following the same pattern as a cpl map of degree 3 (see Figure 6 for details of these patterns)

Proof: Exercise to the reader.

Corollary 3.1:

The singular edges of a cpl map f can be represented as a disjoint union of simple circuits exclusively visiting vertices σ at which $1<rk(f(\sigma))<n$, and simple paths that originate and terminate at vertices σ such that $rk(f(\sigma))=1$ or $rk(f(\sigma))=n$. On such paths and circuits, occurrences of positive and negative singular edges must alternate.

The paths and circuits referred to in Corollary 3.1 will respectively be called the **singular chains** and **singular cycles** of f. With the exception of the trivial cpl maps defined by the free generators of FDL(n), there are no cpl maps for which there exist permutations σ and ρ such that $rk(f(\sigma))=1$ and $rk(f(\rho))=n$. Monotone functions incomparable with the free generators have only singular *cycles*, and these define a partition of S_n. For any monotone function, the minimal rank of the associated cpl map is determined by the size of the shortest prime implicant, and dually. As a further consequence of Proposition 3, it may be seen that knowledge of the set of singular chains and cycles of a cpl map is sufficient to characterise the map completely.

§4 Normal forms and the disposition of singular edges

The nature and disposition of the singular edges can be viewed from another perspective. For the cpl map defined by the boolean function f=g+h in FDL(n), there are three regions in Γ_n:

$R_{g>h}$ - the set of vertices σ at which $rk(f(\sigma))=rk(g(\sigma))>rk(h(\sigma))$:

$R_{h>g}$ - the set of vertices σ at which $rk(f(\sigma))=rk(h(\sigma))>rk(g(\sigma))$:

$R_{g=h}$ - the set of vertices σ at which $rk(f(\sigma))=rk(g(\sigma))=rk(h(\sigma))$.

The singular edges introduced when performing the computational step f=g+h then lie on edges connecting these regions, and consist of positive or zero singularities. In the simplest cases, as when $R_{g>h}$ and $R_{h>g}$ are connected, and $R_{g=h}$ is empty, the new singular edges form a cutset of positive singular edges (though the geometry is in general more complex). As an illustration of this, the geometry of singular edges can be directly related to the standard normal forms for the monotone boolean function f.

If S is a subset of $\{1,2,...,n\}$, then $\Gamma_n(S)$ will denote the set of permutations which have S as prefix. If σ is such a permutation, then $\Gamma_n(S)$ is the connected component that contains σ after all edges associated with a transposition of type $\tau_{|S|}$ have been deleted.

Lemma 4:
Let p be the cpl map of degree n determined by the monotone boolean function

$$x_1 . x_2 . \dots .x_r$$

and let S denote the subset {1,2,...,r} of {1,2,...,n}. Then p has precisely one negative singular edge at each of the r!(n-r)! vertices of $\Gamma_n(S)$, and thus defines a matching on $\Gamma_n(S)$.

If σ is a permutation with prefix S, then p behaves as the function $x_{(r-2).\sigma}.x_{(r-1).\sigma}.x_{r.\sigma}$ on $T_{r-1}\sigma$, as $x_{(r-1).\sigma}.x_{r.\sigma}$ on $K_{i\,r-1}\sigma$, and as $x_{r.\sigma}$ on $T_j\sigma$ and $K_{ij}\sigma$ for $j \neq r-1$.

Proposition 5:
Suppose that p and S are as in Lemma 4, and that f is a monotone boolean function with p as a prime implicant. Then f behaves as p on $\Gamma_n(S)$.

The set of permutations at which f and p have the same value defines a connected subgraph of Γ_n that contains no positive singularity: every permutation σ such that $f(\sigma)=p(\sigma)$ is connected to $\Gamma_n(S)$ by a path consisting of zero or non-singular edges.

An edge (σ,σ'), where $f(\sigma)=p(\sigma)$ and $f(\sigma') \neq p(\sigma')$, is either a positive or zero singular edge.

Proof: Exercise to the reader.

Proposition 5 is illustrated in Figure 7, which depicts the connected subgraphs of the form $\Gamma_n(S)$ for the functions threshold_2 and $x_1x_2+x_2x_3+x_1x_4$ in FDL(4). (Figure 7 hints at a relationship between formulae for f in FDL(n), and partitions of Γ_n defined by cutsets of singular edges: cf §8 Proposition 9 below.)

It is known that a monotone boolean function g is zero replaceable for the purposes of computing f if and only if no prime implicant for f is an implicant for g. This condition may be interpreted as asserting: if p is a prime implicant for f, and p and S are related as in Lemma 4, then the rank of g is greater than the rank of f at all permutations in $\Gamma_n(S)$. Note that this condition is in general weaker than requiring that the rank of g exceeds that of f at all permutations in Γ_n. For the latter condition to hold, it is necessary that, for every prime implicant p′ of g, the sets of indices associated with prime implicants of f that are also implicants of p′ should have empty intersection.

§5 Cycles of singular edges and relations in S_n

Every cycle of singular edges corresponds to a relation in S_n defined by a product of transpositions in which each consecutive pair has adjacent indices. For instance, the monotone boolean function $x_1x_2+x_2x_3x_4+x_1x_4$ in FDL(4) gives rise to a cpl map in which there are two cycles, one of length 6 corresponding to the relation:

$$\tau_1\tau_2\tau_1\tau_2\tau_1\tau_2 = (\tau_1\tau_2)^3 = 1$$

and one of length 18 corresponding to the relation:

$$\tau_1\tau_2\tau_3\tau_2\tau_1\tau_2\tau_3\tau_2\tau_3\tau_2\tau_1\tau_2\tau_1\tau_2\tau_3\tau_2\tau_3\tau_2 = 1.$$

It can be shown that the relations which correspond to singular cycles have a special form: they can be expressed as concatenations of products of transpositions in which increasing and decreasing sequences of indices alternate (cf Figure 8). Explicitly, if

$$a < b < c < \dots. < x < y < z \qquad (\$)$$

is a increasing chain of indices, there is an associated product of transpositions

$$\tau_a\tau_{a+1} \dots \tau_z\tau_{z-1} \dots \tau_b\tau_{b+1} \dots \tau_y\tau_{y-1} \dots \tau_c\tau_{c+1} \dots \tau_x\tau_{x-1} \dots \dots \tau_m \qquad (*)$$

which can be expressed in the form $\tau_a \pi \tau_m$. The relations associated with singular cycles are then concatenations of **alternating sequences** of the form $\tau_a \pi \tau_m \pi^R$, where $\tau_a \pi \tau_m$ has the form (*), and π^R denotes the reversal of the segment π. The special notation [a z b y c x ...] is adopted for the alternating sequence derived from the chain ($) in this fashion. Using this notation, the specimen relations defined above can be represented as

$$[1\ 2]^3 \text{ and } [1\ 3]\,[1\ 3\ 2]\,[1\ 2]\,[1\ 3\ 2]$$

respectively. There are then simple ways to calculate the length of a cycle and the number of local maxima (points at which the sequence of indices changes direction from increasing to decreasing on the cycle): the total length of the cycle is the sum of the lengths of its constituent alternating sequences, where

$$\text{length of } [a_1\ a_2 \ldots a_n] = 2.\Sigma_{1 \le i \le n-1} |a_i - a_{i+1}|,$$

and each alternating sequence $[a_1\ a_2 \ldots a_n]$ contributes n-1 to the number of local maxima. These characteristics of a cycle - its length and *alternation index* - are useful as a means of partial classification; it will be convenient to refer to the specimen cycles above as the 6_3 and 18_6 cycles respectively, for example. This distinguishes the singular cycle of length 18 above from the 18_5 cycle [1 3 2][1 4 2 3] which corresponds to a 5 variable function. (It is of incidental interest to observe that the number of variables required, the length and the alternation index of a cycle can be the same for different cycles cf the two 60_{10} cycles catalogued in the Appendix.)

No necessary and sufficient conditions for a relation in Γ_n to correspond to a singular cycle have yet been derived. By duality, any such cyclic relation must be expressible both as a concatenation of alternating sequences, and as a concatenation of dual alternating sequences, defined in a manner very similar to that described above from a decreasing chain of indices

$$a > b > c > .... > x > y > z.$$

(The specimen relations above admit representations $[2\ 1]^3$ and [3 1][3 2][3 1 2][3 2][3 1] of this form - cf Figure 8.) This restriction on relations - the W-M condition - eliminates many possible products of transpositions from consideration, but is not sufficient to characterise those associated with singular cycles. For instance, the cyclic relation

$$([1\ 4\ 2][1\ 4\ 2\ 3][1\ 4])^5 = 1$$

- which can also be represented as $([4\ 1][4\ 2\ 3][4\ 1]^2[4\ 2])^5 = 1$, and thus necessarily satisfies the W-M condition - corresponds to no cycle of singular edges. In fact, there is a unique 5 variable cpl map for which the sequence [142][1423] occurs in a singular cycle, viz:

$$x_1 x_2 + x_1 x_3 x_4 + x_1 x_3 x_5 + x_2 x_3 x_5$$

which has the singular 36_9 cycle [1 4 2][1 4 2 3][1 4 3][1 3 2] at the permutation 12345. (A simpler similar example is provided by the relation $([1\ 3][1\ 3\ 2])^3 = 1$.)

It is evident that the sequence of transpositions described in abbreviated form by

$$[a\ z\ b\ y\ c\ x \ldots]\ \tau_a$$

is palindromic, so that singular cycles are built up from palindromic segments. Many singular cycles may also be identified as palindromes; for instance, the 18_6 cycle introduced above is palindromic, as may be seen from its dual representation as [3 1][3 2][3 1 2][3 2][3 1]. Not all singular cycles are palindromes however: see the 36_9 cycle above (cf the discussion of circuit building and the structure of singular cycles in §9 below).

From the abbreviated notation for singular cycles described above, it is easy to identify some features of the behaviour of the associated cpl map on the cycle. Every local maximum corresponds to a positive singularity and dually; all other edges are zero singularities. If the abbreviated expression for a singular cycle contains n distinct indices, the behaviour of the cpl map on this particular singular cycle depends only upon n+1 variables. If m and M respectively denote the minimum and maximum indices which appear in the relation, the behaviour of the cpl map of degree N on the singular cycle is unaltered by setting m-1 variables to 1, and N-M-1 variables to 0 (viz those variables whose indices have fixed rank < m or > M throughout the cycle). There may also be variables whose indices have rank lying within the range [m,M] throughout the cycle which at no point designate the value of the cpl map: these can be detected as corresponding to indices in the range [m,M] which do not appear in the abbreviated notation. The singular cycles which arise in this way are essentially trivial variants of smaller cycles, associated with cpl maps of lower degree. For instance, the relation $[1\ 3]^3 = 1$, which gives rise to a singular cycle of length 12 and alternation index 3 is essentially a trivial variant of the 6_3 cycle $[1\ 2]^3$ into which 6 zero singular edges have been interpolated. Such variants are of little interest, and do not merit independent consideration.

A simple example will be used to illustrate the above discussion. For the 5 input monotone boolean function $f=x_1x_2+x_2x_4+x_1x_4+x_1x_3\,x_5$, the associated cpl map has a set of singular cycles which can be classified as follows:

a) Singular cycles which incorporate a permutation ending in 3 or 5

All permutations in such cycles have the same final index. For this reason, the behaviour of f on such cycles is the same as that of the threshold function $x_1x_2+x_2x_4+x_1x_4$ which is the result of setting the variable x_3 or x_5 to 0. The permutations 12435, 31245, 12453 and 51243 in which the indices 1, 2 and 4 appear contiguously lie on distinct singular 6_3 cycles, and the permutations 12345 and 12543 lie on distinct singular 12_3 cycles which are trivial variants of the 6_3 cycles.

b) Singular cycles which contain a permutation beginning with 3 or 5

All permutations in such cycles have the same initial index. For this reason, the behaviour of f on such cycles is the same (up to symmetry) as that of the function $x_1x_2+x_2x_4+x_1x_4+x_1x_3$ which is the result of setting the variable x_5 to 1. The permutation 51234 lies on an 18_6 cycle

$$[1\ 3][1\ 2][1\ 3\ 2][1\ 2][1\ 3]$$

as also (by symmetry) does the permutation 31254.

c) All permutations not beginning or ending in 3 or 5 form a singular 36_8 cycle.

(For another perspective on this classification, see the discussion following §8 Proposition 9.)

A catalogue of relations which has been discovered so far is given in an Appendix; this is not intended to be an exhaustive list of small relations, though it probably contains many of these. It is clear that each relation has a dual obtained by expressing the relation as a concatenation of dual alternating sequences, and relabelling the sequence of indices 1,2,...,n as n,n-1,...,1. Under this duality the cycle 6_3 is self dual, whilst the 18_6 cycle

$$[1\ 3][1\ 3\ 2][1\ 2][1\ 3\ 2] = [3\ 1][3\ 2][3\ 1\ 2][3\ 2][3\ 1]$$

has as dual [1 3][1 2][1 3 2][1 2][1 3], which is associated with the dual of the monotone boolean function $x_1x_2+x_2x_3x_4+x_1x_4$, viz: $x_1x_2+x_2x_4+x_1x_4+x_1x_3$. The configuration (see §8 below) associated with the dual of a relation is derived by transposing rows and columns.

§6 Elementary group theoretic properties of alternating sequences and associated relations

Methods for recognising and synthesising the relations corresponding to singular cycles would potentially be of interest. Though it is relatively easy to discover new relations by serendipity, or by performing simple transformations on monotone boolean functions (see §8 below), more systematic procedures for generation and analysis are probably needed in order to determine the computational relevance of singular cycles for monotone boolean function complexity. What is the asymptotic length and alternation index of the most complicated singular cycle which can be realised with n variables, for instance? In this section, alternating sequences are viewed from a group theoretic perspective as defining explicit permutations, and some elementary properties are observed. One motivating idea is the development of group theoretic methods for constructing singular cycles which would lead directly to the synthesis of monotone boolean functions with interesting computational characteristics.

Lemma 6:

The permutation represented by the alternating sequence [1 a b ... y z] is (1 2)(y+1 z+1). Equivalently:

the permutation represented by the alternating sequence [1 k] ($k \geq 2$) is the 3-cycle (1 k+1 2), and the permutation represented by the alternating sequence [1 a_1 a_2 ... a_n a_{n+1}] ($n \geq 2$) is the double transposition (1 2) (a_n+1 a_{n+1}+1).

Proof: Exercise to the reader.

Lemma 6 shows that alternating sequences necessarily represent even permutations. In fact, it can be shown that the subgroup of S_n generated by alternating sequences is the alternating group A_n. Indeed, for $n \geq 3$, A_n is generated by the alternating sequences of the form [1 k]. To see this, observe that

$$(1\ a\ 2)(1\ b\ 2)^2 = (1\ a\ b) \quad \text{and} \quad (1\ a\ b)(1\ c\ a) = (a\ b\ c)$$

and the set of 3-cycles is known to generate A_n (see eg Hall [6] p61).

For n=4, the relations satisfied by products of alternating sequences can conveniently be determined from Figure 9: the Cayley diagram for A_4 as generated by the basic alternating sequences [1 2] and [1 3]. As depicted in Figure 9, the alternating sequence [1 3 2] is represented by the permutation of order 2 that corresponds to the two products:

$$[1\ 2]^2[1\ 3][1\ 2]^2 \equiv [1\ 2][1\ 3]^2[1\ 2].$$

The non-trivial relations that arise as singular cycles of monotone boolean functions in 4 variables are:

12_4	$[1\ 3\ 2]^2$ - dual of $([1\ 3][1\ 2])^2$
18_6	$[1\ 3][1\ 3\ 2][1\ 2][1\ 3\ 2]$ - dual of $[1\ 2][1\ 3]^2[1\ 2][1\ 3\ 2]$
24_8	$([1\ 3\ 2][1\ 2][1\ 3])^2$ - self-dual
24_8	$([1\ 2][1\ 3\ 2])^3$ - self-dual

As remarked above, not all the group-theoretic relations between [1 2],[1 3] and [1 3 2] are associated with singular cycles: the relation $([1\ 3][1\ 3\ 2])^2$, for instance, traces a circuit of Γ_4 which is not simple; it in fact incorporates the 12_4 cycle $[1\ 3\ 2]^2$.

Conjecture: The relations between alternating sequences which correspond to singular cycles are precisely those which satisfy the WM-condition and define simple circuits of Γ_n.

It is of some interest to determine generic classes of relations that arise as singular cycles. Lemma 6 provides the basis for identifying such classes. A simple example of such a class is defined by the sequence

$$12_4 : [1\ 3\ 2]^2,\ 24_6 : [1\ 4\ 2\ 3]^2,\ 40_8 : [1\ 5\ 2\ 4\ 3]^2,\ 60_{10} : [1\ 6\ 2\ 5\ 3\ 4]^2, ...$$

of singular cycles for monotone functions in 4, 5, 6, 7, ... variables, of which the generic instance has length 2(n-1)(n-2) and alternation index 2(n-2). Singular cycles of each type arise in connection with functions lying within specified intervals, but a natural choice for a representative function with the singular cycle $2(n-1)(n-2)_{2(n-2)}$ is $\text{alt}_\Sigma(x_1, x_3, x_4,..., x_{n-1}).\text{alt}_\Sigma(x_2, x_3, x_4,..., x_{n-1})$, where alt_Σ is the alternating sum function to be defined in §7 below.

For the present, it is hard to establish generic relations of the above type; more effective proof techniques are required to justify the association of relations and singular cycles rigorously. It seems probable that there are many such classes however.

Conjecture 1:
If r denotes the integer part of n/2, then the relations

$$([1\ r][1\ n\text{-}1\ 2\ n\text{-}2\ ....\ r\ r{+}1])^3 \text{ (n odd) and } ([1\ r][1\ n\text{-}1\ 2\ n\text{-}2\ ....\ r{+}1\ r])^3 \text{ (n even)}$$

of type $3n(n-2)_{3(n-1)}$ can be realised as singular cycles of functions in FDL(n). For n=4,6,8,... appropriate functions of a generic form are:

$$x_2(x_1+x_3+x_4)+ x_1x_3x_4$$
$$x_2(x_6+x_3(x_1+x_4+x_5))+ x_1x_4x_5$$
$$x_2(x_6+ x_3(x_7+ x_4(x_1+x_5+x_6))+ x_1x_5x_6$$

....

Conjecture 2:
Every relation of the form [1 ... y z][1 ... y z] or [1 ... y z][1 ... z y] can be realised as a singular cycle. An appropriate generic form for functions to realise these cycles is:

$$\text{alt}_\Sigma(...).\text{alt}_\Sigma(...), \text{ or dually: } \text{alt}_\Pi(...)+\text{alt}_\Pi(...)$$

for suitably chosen alternating sums or products in which selected variables have been assigned to 0 or 1. For instance, the relations:

18_5 : [1 3 2][1 4 2 3],
28_6 : [1 4 3][1 5 2 4 3],
38_7 : [1 4 3][1 6 2 5 3 4],
52_8 : [1 5 4][1 7 2 6 3 5 4],
....

form a generic class of relations of type $((n-1)(n-2)+2(r+1))_n$, for which a suitable representation as a sum of alternating products is

$$x_2(x_1+x_n(x_3+x_{n-1}(\ x_4+.....)\ +\ x_1x_rx_{r+1}.$$

The rationale for viewing the latter term above as the result of setting variables is best appreciated by referring to the first example of a generic class given above.

It does not appear to be easy to determine the maximum length of the singular cycle that can be obtained using n generators. It would be of interest to know whether there are generic classes of cycles whose length grows faster than quadratically with n for instance. The elementary observations that follow indicate some of the features which constrain the potential length of cycles.

On any singular cycle (without redundancy) on the inputs {1,2,...,n}, there are permutations at which the rank of the associated cpl map is 2 and n-1. Such a map necessarily corresponds to a monotone boolean function f which has at least one prime implicant and at least one prime clause of length 2. Let P_2 and Q_2 respectively denote the set of the prime implicants and prime clauses of f of length 2. It will be convenient to represent P_2 and Q_2 as sets of edges of a graph on the vertex set {1,2,...,n}. In view of the characteristic properties of prime implicants and clauses, every edge of P_2 is incident with every edge of Q_2. Since these implicants appear on a singular cycle, the subgraphs P and Q of the complete graph on the vertex set {1,2,...,n} respectively defined by the edge sets P_2 and Q_2 are connected.

From the above observations, it can be seen that a Hamiltonian singular cycle occurs only if $n\leq 4$. To prove this: consider a singular cycle (without redundancy) on the inputs {1,2,...,n}, let P and Q be as defined above, and suppose (wlog) that $q=\langle 1,2\rangle$ is an edge in Q. All edges in P are then incident with 1 or 2, and there must be edges $p_1=\langle 1,i\rangle$ and $p_2=\langle 2,j\rangle$ in P, where $1\leq i\leq j\leq n$. If $n\geq 4$, i and j can clearly be chosen to be distinct indices. Moreover, if n>4, there must also be an index k distinct from 1,2,i and j, and - by connectivity of P - it can be assumed that (relabelling vertices 1 and 2 if necessary) $p_3=\langle 1,k\rangle$ is also in P. Any edge in Q is incident with both p_1 and p_3, and necessarily has 1 as a vertex, since $\langle i,k\rangle$ is disjoint from p_2. There can therefore be no edge of Q incident with i or k, from which it follows that no permutation σ such that $n.\sigma$ = i or k appears on the singular cycle.

§7 Alternating sums and products

The existence of a particular singular cycle indicates that a monotone boolean function specialises to a function of a particular form under an appropriate assignment of boolean variables to selected variables. Ideally, it would be desirable to find a classification of singular cycles, and to characterise the interval of monotone boolean functions associated with each singular cycle. If this proved possible, a further objective might be to interpret the set of singular cycles associated with a particular monotone boolean function as representing a canonical decomposition of some kind. In view of the omnipresence of alternating sequences in singular cycles, it seems probable that a particularly simple class of monotone boolean functions - the alternating sums and products - will be of direct relevance in making connections between singular cycles and monotone boolean functions (cf the examples of generic cycles in §6).

The *alternating sum* $\mathrm{alt}_\Sigma(x_1, x_2, x_3, x_4,\ldots, x_n)$ denotes the monotone boolean function

$$x_1 + x_2 . (x_3 + x_4 . (x_5 + \ldots)\ldots)$$

and the *alternating product* $\mathrm{alt}_\Pi(x_1, x_2, x_3, x_4,\ldots, x_n)$ the dual function

$$x_1 . (x_2 + x_3 . (x_4 + x_5 .(\ldots)\ldots)).$$

Lemma 7:

The alternating sum $\mathrm{alt}_\Sigma(x_1, x_2, x_3, x_4,\ldots, x_n)$ has the conjunctive normal form:

$(x_1 + x_2).(x_1 + x_3 + x_4). \ldots .(x_1 + x_3 + \ldots + x_{n-1} + x_n)$ (n even)

$(x_1 + x_2).(x_1 + x_3 + x_4). \ldots .(x_1 + x_3 + \ldots + x_{n-2} + x_{n-1}).(x_1 + x_3 + \ldots + x_{n-2} + x_n)$ (n odd)

and the disjunctive normal form:

$x_1 + x_2.x_3 + x_2.x_4.x_5 + \ldots + x_2.x_4. \ldots .x_{n-2}.x_{n-1} + x_2.x_4. \ldots .x_{n-2}.x_n$ (n even)

$x_1 + x_2.x_3 + x_2.x_4.x_5 + \ldots + x_2.x_4. \ldots .x_{n-1}.x_n$ (n odd)

(A dual result - derived by interchanging '+' and '.' - applies to the alternating product.)

Proof: Induction on n - exercise to the reader.

Alternating sums and products are naturally associated with alternating sequences of products of transpositions. To describe the correspondence between them, it will be convenient to adopt a notation that distinguishes between an alternating sequence occurring as a segment within a singular cycle, and an alternating sequence occurring as a singular chain. The rationale for this is that the (extremal) rank attained by a cpl map on a singular chain differs from that on a singular cycle; moreover there is a need for an appropriate notation to reflect the occurrence of sequences such as

[13][12][13]

as singular chains. The solution is to introduce the symbols 0 and n to designate the points at which the a cpl map of degree n attains extremal rank on a singular chain. With this convention, the above sequence - that otherwise appears anomalous - is denoted by [0132].

Lemma 8: The singular chain [0 a b c .... z] appears at the permutation 12 ... N for the cpl map

$$\mathrm{alt}_\Sigma(x_1, x_{a+1}, x_{b+1}, x_{c+1},\ldots, x_{z+1})$$

Proof: Exercise to the reader.

Lemmas 6 and 8 together make it possible to find boolean functions that exhibit a specific singular cycle (given as products of transpositions defining an appropriate relation in S_n) at a given permutation. For instance: a monotone boolean function that gives rise to the 18_6 cycle [132][1423] at the permutation 12345 can be identifed as

$$x_2.(x_1 + x_4.x_3) + x_1.(x_2 + x_5.(x_4 + x_3)).$$

(To derive this, observe that the required cycle can be expressed as

$$\tau_1(\tau_2\tau_3\tau_2\tau_3\tau_2)\tau_1(\tau_2\tau_3\tau_4\tau_3\tau_2\tau_3\tau_2\tau_3\tau_4\tau_3\tau_2)\tau_1,$$

with the result that an appropriate function traces a [021] cycle in the context $x_2=1$ at the permutation 21345, and a [0312] cycle in the context $x_1=1$ at the permutation 12435.)

The above observations suggest that the decomposition into singular cycles associated with a cpl map can be interpreted as representing the corresponding boolean function in terms of alternating sums involving the generators. In this connection, it is of incidental interest to note that the functions of the form $alt_\Sigma(x_{1.\sigma}, x_{2.\sigma}, x_{3.\sigma},..., x_{n.\sigma})$ and $alt_\Pi(x_{1.\sigma}, x_{2.\sigma}, x_{3.\sigma},..., x_{n.\sigma})$, where σ ranges over S_n, generate all functions in FDL(n) bar the free generators. (To see this, observe that the product of all alternating sums of the form

$$alt_\Pi(x_{1.\sigma}, x_{2.\sigma}, x_{3.\sigma},..., x_{n.\sigma})$$

where $1.\sigma=i$ is the unique element that covers x_i in FDL(n). This element can play the role of x_i in any non-trivial disjunction in which the element x_i enters a computation, and dually.)

§8 Singular cycles and configurations of prime implicants and clauses

The theory of computational equivalence for monotone boolean functions shows that the computational usefulness of a monotone boolean function g for the purpose of computing a given function f depends entirely upon its relation to the prime implicants and clauses of f. The following proposition is a consequence:

Proposition 9:

Suppose that $f\in$ FDL(n), and let R_f be the rectangular array whose rows are indexed by the prime implicants of f, whose columns are indexed by the prime clauses, and whose (i,j)-th entry is the set of indices of generators appearing in the intersection of the i-th prime implicant and the j-th prime clause.

Then

(i) a formula that represents f corresponds to a partition of R_f into blocks, each of which is a generalised subrectangle in which all entries have an index in common, having the property that R_f can be built up from its constituent blocks by repeatedly gluing together blocks having representatives in identical rows or columns. The number of blocks in such a partition is the number of inputs to the formula.

(ii) a circuit for f corresponds to a sequence of operations on R_f each of which selects a pair of indices (x,y) appearing in the same row (respectively column) of R_f and adjoins a new index to every entry containing x or y in that row (respectively column) until such time as all entries in R_f have an index in common. The number of such operations required is the size of the circuit.

Proof:

(i) A formula for the boolean function f can be interpreted as a circuit in the form of a tree whose leaves are labelled by input variables, and whose internal nodes correspond to **and** and **or** gates. Following the notation of [2], let P_f (respectively Q_f) denote the set of prime implicants (respectively clauses) of f. To each node γ of this tree, attach a code $<\alpha,\beta>$, where α (resp. β) is a binary string of length $|P_f|$ (resp. $|Q_f|$) such that α_i (resp. β_i) is 1 if and only if the function computed by γ has α_i as an implicant (resp. β_i as a clause).

To derive a partition of R_f, a simple procedure is first applied to each node in the tree on a top-down, breadth first basis. The effect is in general to systematically alter the codes for all nodes with the exception of the root. Suppose that all ancestors of the gate γ have been processed, and that γ is an **and** gate with the modified code $<\alpha',\beta'>$. All 1's in α' must be common to both gates that are direct inputs to γ; the codes of both these input gates are altered to take the form $<\alpha',*>$. Each 1 that appears in β' must appear in at least one of the input gates to γ; select a representative for each such 1 from one of the gates only, and modify the codes for the input gates to the form

$<\alpha', \beta''>$, where the only 1's in β'' are representatives for 1's in γ.

The blocks of the required partition are then defined by the generalised rectangles in R_f whose rows and columns are respectively indexed by α_ι and β_ι, where $<\alpha_\iota, \beta_\iota>$ is the code allocated to the leaf node associated with input gate ι.

(ii) The computational equivalence class of $g \in FDL(n)$ relative to f is entirely determined by its relationship to the prime clauses and prime implicants of f. The effect of the procedure described in (ii) is simply to introduce a new index to represent this computational equivalence class explicitly in the array R_f. For instance, if g=h.k, and p and q respectively represent a prime clause and a prime implicant of f, then $p \le g$ iff $p \le h$ and $p \le k$, whilst $g \le q$ iff $h \le q$ or $k \le q$. Introducing g then corresponds to adjoining the new index representing g to each entry in R_f in which at least one of h or k is represented, and such that both h and k are represented within the corresponding row. The function f is uniquely represented by an index that appears in all entries of the array R_f.

♦

Proposition 9 indicates a strong connection between configurations of prime implicants and clauses and the computational characteristics of a monotone boolean function; a theme to be developed elsewhere [3]. It can be shown, for instance, that the array R_f consists exclusively of singletons if and only if f can be expressed as a formula without duplication of the inputs. (For simple illustrative examples of Proposition 9, see Figure 10.)

For non-trivial functions f, analysis of the array R_f appears to be difficult, but it can be shown that the presence of particular singular cycles is connected with features of R_f. A necessary and sufficient condition for $f \in FDL(n)$ to possess a particular singular cycle at a given permutation is that it should belong to a particular interval in FDL(n). In fact, the form of the singular cycle is a guarantee that f possesses a particular set of implicants and clauses. These sets can be determined by examining the permutations on the cycle at which the rank of f attains its extremal values; on each such permutation the set of inputs whose rank is at most the rank of f determines a required implicant, and the set of inputs whose rank is at least the rank of f a required clause. To express this condition in terms of the array R_f, it is necessary to identify the minimal elements in the required sets of implicants and clauses; their presence then corresponds to the existence of a particular pattern of incidence within the array R_f.

To illustrate this, consider the necessary and sufficient conditions for a monotone boolean function f in FDL(5) to have the 18_5 singular cycle [132][1423] at the permutation 12345. Tracing the singular cycle, monitoring the implicants and clauses required to ensure the appropriate extremal points occur, shows that f has the implicants: x_1x_2, $x_2x_3x_4$, $x_1x_3x_5$, $x_1x_4x_5$, and the clauses: $(x_1+x_4+x_5)$, $(x_1+x_3+x_5)$, (x_2+x_5), $(x_2+x_3+x_4)$. The pattern of intersection between required implicants and clauses then defines the 4×4 array:

1	1	2	2
4	3	2	234
15	135	5	3
145	15	5	4

that does not of course correspond directly to a pattern of intersection between prime implicants and prime clauses exhibited by any single function f in the specified interval. An appropriate way to describe the possible configurations of intersections associated with such functions f is to extract from each row (respectively column) any indices which do not appear as singletons, and to record these separately in an additional column (respectively row). These extra row (respectively column) entries correspond to variables that may or may not appear in the implicant (respectively clause) associated with the row (respectively column) - depending on the choice of f. Since the form of the non-singleton entries in the array is dependent on the choice of function f, and is uniquely determined by which of the optional variables appear in prime implicants and prime clauses of f, it is then convenient to omit these entries from the array. With this convention, the pattern of intersection that is associated with the 18_5 singular cycle [132][1423] at the permutation 12345 is represented by the configuration:

$$\begin{array}{ccccc} & 1 & 1 & 2 & 2 \\ & 4 & 3 & 2 & * \\ \mathit{1} & * & * & 5 & 3 \\ \mathit{1} & * & * & 5 & 4 \\ & \mathit{5} & \mathit{5} & & \end{array}$$

Note that for the particular monotone boolean function

$$x_2.(\, x_1 + x_4.x_3\,) + x_1.(\, x_2 + x_5.(x_4 + x_3\,))$$

that was derived above, a specific form of this configuration arises from the prime implicants

$$x_1x_2,\ x_2x_3x_4,\ x_1x_3x_5,\ x_1x_4x_5$$

and the prime clauses (x_1+x_4), (x_1+x_3), (x_2+x_5) and $(x_2+x_3+x_4)$, viz:

$$\begin{array}{ccccc} & 1 & 1 & 2 & 2 \\ & 4 & 3 & 2 & * \\ \mathit{1} & 1 & * & 5 & 3 \\ \mathit{1} & * & 1 & 5 & 4 \end{array}$$

The classification of configurations is a variant of the problem of classifying intervals of monotone boolean functions associated with singular cycles referred to above (cf Figure 11, which depicts the lattice interval in FDL(5) associated with the presence of a 20_6 singular cycle). It is easy to see that any array that defines such a configuration contains at least two distinct indices in every row and column. Specialisation of a function by setting an input x_i appearing within a configuration to 0 (respectively 1) can be interpreted as "deleting all rows (respectively columns) containing index i". If the effect of such an operation is to produce a subconfiguration ie an array with at least two distinct indices in every row and column, this indicates the presence of additional singular cycles. For instance, the 36_8 cycle $([132][14]^2)^2$ has the characteristic configuration:

$$\begin{array}{cccc} 2 & 2 & 1 & * \\ 4 & 4 & * & 1 \\ * & * & 4 & 2 \\ 3 & 5 & 1 & 1 \end{array}$$

that yields subconfigurations on the assignments $x_3=0$, $x_5=0$, $x_3=1$, $x_5=1$. By referring to the catalogue in the Appendix, the complete classification of singular cycles for the function

$$x_1x_2+x_2x_4+x_1x_4+x_1x_3\,x_5$$

- as specified in detail in a previous section - can be derived.

Reference to the catalogue in the Appendix shows that interdependence of singular cycles is a common occurrence. By the arguments above, the presence of the 28_6 cycle $([142][13])^2$ guarantees the presence of a 20_6 cycle $([143][12])^2$, for instance. Figure 12 shows how this relationship between cycles can be interpreted in the Cayley diagram Γ_5.

§9 Singular cycles and the complexity of monotone boolean functions

Monotone boolean function complexity provided the original motivation for the study of singular cycles. It might be hoped that functions with large circuit size would necessarily possess singular cycles of a particularly complicated form. Such a hope is confounded by slice functions, most of which are known to require large circuits by counting arguments [9], but that can have singular cycles associated only with cpl maps of degree at most 4. Nor is the number of singular cycles an appropriate indication of complexity, since threshold functions and slice functions have very similar characteristics. It would be interesting to find bounds on the length and alternation indices of singular cycles associated with cpl maps of degree n, and on the complexity of the functions needed to realise these particular cycles, but it is not clear at present whether this can offer a method for constructively demonstrating non-trivial lower bounds on circuit size. Perhaps it is more reasonable to expect that the complexity of realising an individual singular cycle will prove to be small in general, and that the complexity of a function is associated with "simultaneously realising many different cycles", as must be the case for slice functions. Such arguments would necessarily have to consider not only the form of the singular cycles associated with a given cpl map, but how inputs must be assigned in order to realise these cycles.

Attempts to apply the above theory to proving lower bounds on circuit complexity have yet to produce interesting concrete results. One possible approach is to consider the implications for circuit size of computing a function that can be specialised to yield many different alternating sums and products (cf [5]). It is easy to see that an alternating sum or product of input variables admits one optimal computation, and that the order in which inputs are introduced in this computation is completely determined. Informally, it seems plausible that functions that could be specialised to yield alternating sums and products associated with many different variable orderings should be hard to realise. The work on computational equivalence and replaceability described in [2] was originally motivated by the study of such issues.

A standard technique, first introduced by Schnorr [8], but yet to be successfully applied to proving non-trivial lower bounds for monotone boolean functions, aims to find invariants that can be shown to attain large values for certain functions, but that cannot increase very greatly under the operations that correspond to the construction of a boolean circuit or formula. For the reasons explained above, simple invariants associated with single cycles can only have limited significance here, but consideration of how circuit building operations can affect the geometry of singular cycles has some intrinsic interest. One attraction of exploring boolean circuit building operations in the context of cpl maps is that the **and-or** duality takes a very simple form, and the geometry of singular cycles respects this duality. (Schnorr's unsuccessful attempts to extend the arguments in [8] to monotone boolean functions were based upon combinatorial features of the prime implicants.) It is thus enough to consider two types of operation: the mapping FDL(n) $\rightarrow$ FDL(n+1) defined by replacing the input x_n by $x_n.x_{n+1}$, and the mapping FDL(n+1) $\rightarrow$ FDL(n) defined by identifying the inputs x_n and x_{n+1}.

Operation 1: replacing the input x_n by $x_n.x_{n+1}$

Suppose that $f \in$ FDL(n), that $F \in$ FDL(n+1), and that

$$F(x_1, x_2, ..., x_n, x_{n+1}) \equiv f(x_1, x_2, ..., x_{n-1}, x_n.x_{n+1}).$$

It will be convenient to assume that f - and thus F - is incomparable with any generator x_i. (The derivation of similar results for singular chains is left as an exercise to the reader.) The singular cycles of F can then be inferred from those for f. Suppose that C is a cycle for F. There are essentially two possible cases to consider:

Case 1: C contains no edge on which the indices n and n+1 are transposed

In this case, the order in which n and n+1 appears in permutations on C is fixed. For each $\sigma \in C$, let σ' be the permutation of {1,2,...,n} obtained by deleting $i.\sigma$ from σ, where i is the smaller of the two indices for which $i.\sigma$ = n or n+1. The result of this operation is to transform C into a cycle c for f; moreover C is essentially equivalent to the singular cycle c: it is either identical in form, or differs by the interpolation of zero singular edges. In effect, c is derived from C by eliminating the variable $x_{i.\sigma}$, where the behaviour of F on C is independent of $x_{i.\sigma}$.

Case 2: C contains an edge on which the indices n and n+1 are transposed

In this case, C contains permutations σ, $\sigma' = \tau_r \sigma$ representable (as sequence of indices) as:

$$\sigma \equiv \alpha\ n\ n{+}1\ j\ \beta \text{ and } \sigma' \equiv \alpha\ n{+}1\ n\ j\ \beta$$

where α and β are sequences of indices, α has length r-1, and $rk(F(\sigma))=rk(F(\sigma')) = r+1$. Let ρ and ρ' be the permutations of {1,2,...,n} defined by the sequences:

$$\rho \equiv \alpha\ n\ j\ \beta \text{ and } \rho' \equiv \alpha\ j\ n\ \beta.$$

Necessarily $f(\rho)$ = n, and the edge (ρ,ρ') is either zero-singular or positive singular for f. Let c be the singular cycle of f that contains (ρ,ρ'). Consider a traversal of the cycle c that begins at ρ and proceeds in the direction (ρ,ρ'), and let

$$\rho_0=\rho,\ \rho_1=\rho',\ldots,\ \rho_k,\ldots$$

be the sequence of permutations encountered. Suppose that t is the smallest positive index such that

$$f(\rho_t) = n \ \text{ and } \ rk(f(\rho_t)) = r.$$

Such a t necessarily exists, since the permutation that directly precedes ρ_0 on the cycle c necessarily has the index n at rank < r. For the very same reason, $\rho_0 \neq \rho_t$. The form of the cycle C is now readily described. In effect, C has a traversal

$$\sigma_0=\sigma,\ \sigma_1=\sigma',\ldots,\ \sigma_{t'}$$

that mimics the traversal $\rho_0=\rho,\ \rho_1=\rho',\ldots,\ \rho_t$ exactly on those edges $(\rho_i,\ \rho_{i+1})$ at which a pair of indices $\leq$ n is transposed, and interpolates a zero singular edge to transpose a pair of indices of th form (n+1,k) whenever this is required. To be more explicit, "mimicing the traversal"

$$\rho_0,\ \rho_1,\ldots,\ \rho_t$$

at $\sigma_0=\sigma$ may lead to a permutation in which the index n+1 separates the two indices (j,k) that are transposed at the next edge of the cycle c, at which point transposition of k and n+1 is required to make j and k adjacent. Because of the nature of the functional dependence of F upon the variable x_{n+1}, this variable is essentially redundant throughout the entire traversal

$$\sigma_0=\sigma,\ \sigma_1=\sigma',\ldots,\ \sigma_{t'}$$

so that the rank of the index n+1 is either constant or oscillates between r and r-1. In particular, the index at rank r-1 in $\sigma_{t'}$ is necessarily n+1. Because of the symmetry between n and n+1, it then follows that C is a singular cycle of the form

$$\sigma_0,\ \sigma_1,\ldots,\ \sigma_{t'},\ \sigma_{t'}\tau,\ldots,\ \sigma_1\tau,\ \sigma_0\tau,\ \sigma_0$$

for which the corresponding relation in S_4 is palindromic. (See Figure 13 for an illustration of how the 18_6 cycle associated with the function $x_1x_2+x_1x_3+x_1x_4+x_2x_4$ is transformed into 40_{10} and 20_6 cycles via the replacement of x_4 by $x_4.x_5$.)

Operation 2: identifying the inputs x_n and x_{n+1}

Suppose that $F\in FDL(n+1)$, that $f\in FDL(n)$, and that

$$f(x_1, x_2, \ldots, x_n) \equiv F(x_1, x_2, \ldots, x_{n-1}, x_n, x_n).$$

Within the Cayley diagram Γ_{n+1}, consider the set H of permutations in which the indices n and n+1 are adjacent. There is a 2-1 correspondence $\chi: H \to \Gamma_n$, under which η and η' have the same representative in Γ_n when n and n+1 are identified. Note also that each pair (η,η') such that $\chi(\eta)=\chi(\eta')$ defines an edge of Γ_{n+1}. Suppose that (η,η') and (ψ,ψ') are two such edges, and that $\chi(\eta)$ and $\chi(\psi)$ are adjacent in Γ_n. There are then two possible cases to consider:

Case 1: the four permutations η,η',ψ and ψ' define a coset of the form K_{rs}

In this case - without loss of generality - (η,ψ) and (η',ψ') are edges of Γ_{n+1}, and both necessarily have the same characteristic with respect to F: they are either both non-singular, both zero singular, both positive singular or both negative singular. It is then easy to verify that the edge $(\chi(\eta),\chi(\psi))$ in Γ_n also has the same characteristic with respect to f.

Case 2: the four permutations η,η',ψ and ψ' appear in a coset of the form T_r

In this case - without loss of generality - there are then relations

$$\eta=\tau_r\eta',\ \rho\equiv\tau_{r-1}\eta=\tau_r\psi,\ \psi=\tau_{r-1}\psi' \text{ and } \rho'\equiv\tau_{r-1}\eta'=\tau_r\psi',$$

where (η',ρ'), (ρ',ψ'), (η,ρ) and (ρ,ψ) are edges of Γ_{n+1}. If F is non-singular on all these edges, then f is non-singular on the edge $(\chi(\eta),\chi(\psi))$ in Γ_n. The singularities of F on these edges otherwise conform to one of the patterns depicted in Figure 6. In Γ_{n+1}, assign the code +1 to a positive singular edge, -1 to a negative singular edge, and 0 to other edges. A trivial analysis of all possible cases then shows that the sum of the encodings of the edges (η,ρ) and (ρ,ψ) is the same as that of (η',ρ') and (ρ',ψ'). Furthermore, the edge $(\chi(\eta),\chi(\psi))$ in Γ_n is positive singular, negative singular or zero singular according as this sum is +1, -1 or 0.

It may be seen that the effect of identifying inputs on the form of singular cycles depends on the entire disposition of singular edges for the original function; for this reason, it is hard to predict. The possible forms of singular cycles that can arise by replacing an input by an **and** or an **or** of inputs, on the other hand, can be determined from knowledge of the singular cycles originally present. (Contrast Figure 13 with Figure 14, in which the synthesis of a singular 30_8 cycle upon identification of the inputs x_5 and x_6 in the function $x_1x_2x_4+x_1x_3x_5+x_2x_3x_6+x_4x_5x_6$ is illustrated.)

§10 Future developments and potential applications

Future development of the ideas set out above has two aspects: further theoretical investigation and the identification of applications.

From a theoretical perspective, the most significant problem is the classification of singular cycles, since this seems likely to lead to new insights into the form of monotone boolean functions. A number of different approaches to this problem of classification have been suggested in this paper: for instance, an appropriate characterisation of singular cycles might be expressed in terms of relations in Γ_n, or in terms of intervals in FDL(n). From a group theoretic perspective, connections with research into Bruhat orderings of Coxeter groups (cf [4]) seem plausible, since these in particular relate the structure of relations in symmetric groups to abstract simplicial complexes, and these can in turn be related to the theory of distributive lattices [2]. (No direct links have yet been established.) From a lattice theoretic perspective, it should be stressed that the structure of free distributive lattices is by no means fully understood; as a trivial indication, the computation of the size of FDL(n) has only been achieved for $n\leq 7$. It may be conjectured that the partition of Γ_n associated with a monotone boolean function in FDL(n) corresponds in some way to a normal form whose components are determined by the singular cycles. The potential significance of possessing particular singular cycles still requires clarification, however. The geometric content of singular cycles is perhaps best viewed intuitively as reflecting the 1-dimensional structure of a monotone boolean function; perhaps it will be necessary to not only to understand this structure but to find techniques for higher dimensional analysis (cf Figure 14).

The successful identification of applications might be useful in suggesting directions for further theoretical development. The issues raised by complexity analysis have already been discussed in §9, but there are other potential applications to explore. Computational geometry is one area in which the geometric model of monotone boolean functions may be relevant (cf [1]). It is clear for instance that cpl maps could be useful as a way of abstractly representing continous geometric structures other than piecewise linear maps. Combinatorial abstraction of this kind can be useful as a basis for subsequent analysis: it would be interesting to see whether (for example) the work of Sharir and others on bounding the number of components in the lower envelope of a set of functions [7] can profitably be reinterpreted in the above context. There are more tenuous connections with issues raised by concurrent programming. A cpl map can be viewed as recording the index of the input variable that causes a monotone boolean function to be switched on when the input variables are themselves switched on in a specified order (cf the proof of Lemma 1 in §2). The consideration of such behaviour of a function may be relevant in a context where the function serves as a guard, and the input variables are under the control of independent processes operating concurrently. There may also be prospects for efficient testing of logical circuits based upon the patterns of behaviour observed on singular cycles, though such methods would require a better understanding of the significance of the singular cycle structure.

Acknowledgements

I am grateful to Mike Paterson for motivating ideas and some helpful discussions.

I also wish to thank Simon Giles for his programming assistance, and Alun Morris for drawing my attention to reference [4].

References

[1] W M Beynon, *Combinatorial aspects of piecewise linear maps*, J London Math Soc (2) 7 (1974), 719-727

[2] W M Beynon, *Replaceability and computational equivalence for monotone boolean functions*, Acta Informatica 22, 433-449 (1985)

[3] W M Beynon and M S Paterson (in preparation)

[4] A Bjorner, *Orderings of Coxeter groups*, Proc "Combinatorics and Algebra" Boulder, June 1983, Amer Math Soc 1983

[5] B Commentz-Walter, *Size-depth tradeoff in Boolean formulas*, TB Univ d Saarlandes, AA77-1o (1977)

[6] M Hall, *The theory of groups*, MacMillan 1959

[7] S Hart and M Sharir, *Non-linearity of Davenport-Schinzel sequences and of generalized path compression schemes,* Combinatorica Vol 6 (2), 151-177, 1986

[8] C P Schnorr, *A lower bound on the number of additions in monotone computations*, Theoretical Computer Science 2 (1976), 305-315

[9] I Wegener, *On the complexity of slice functions*, Univ of Frankfurt, Internal Report, 1983

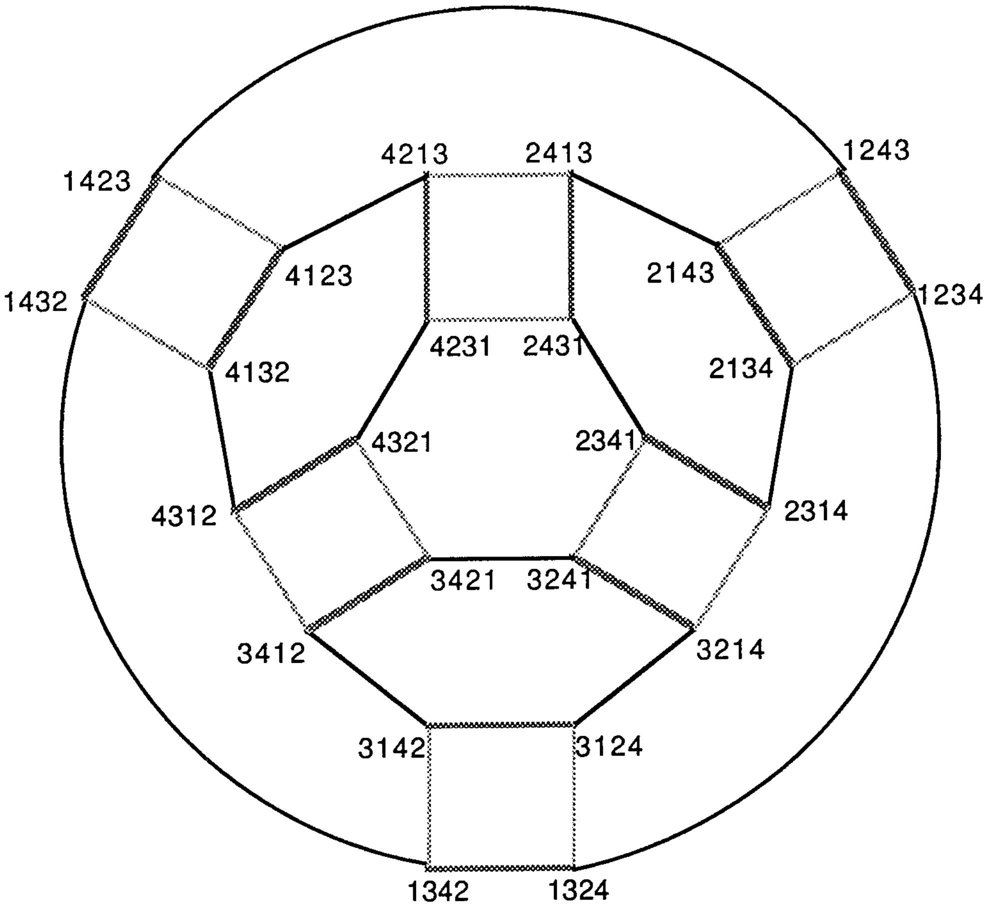


Figure 1: The Cayley diagram Γ_4

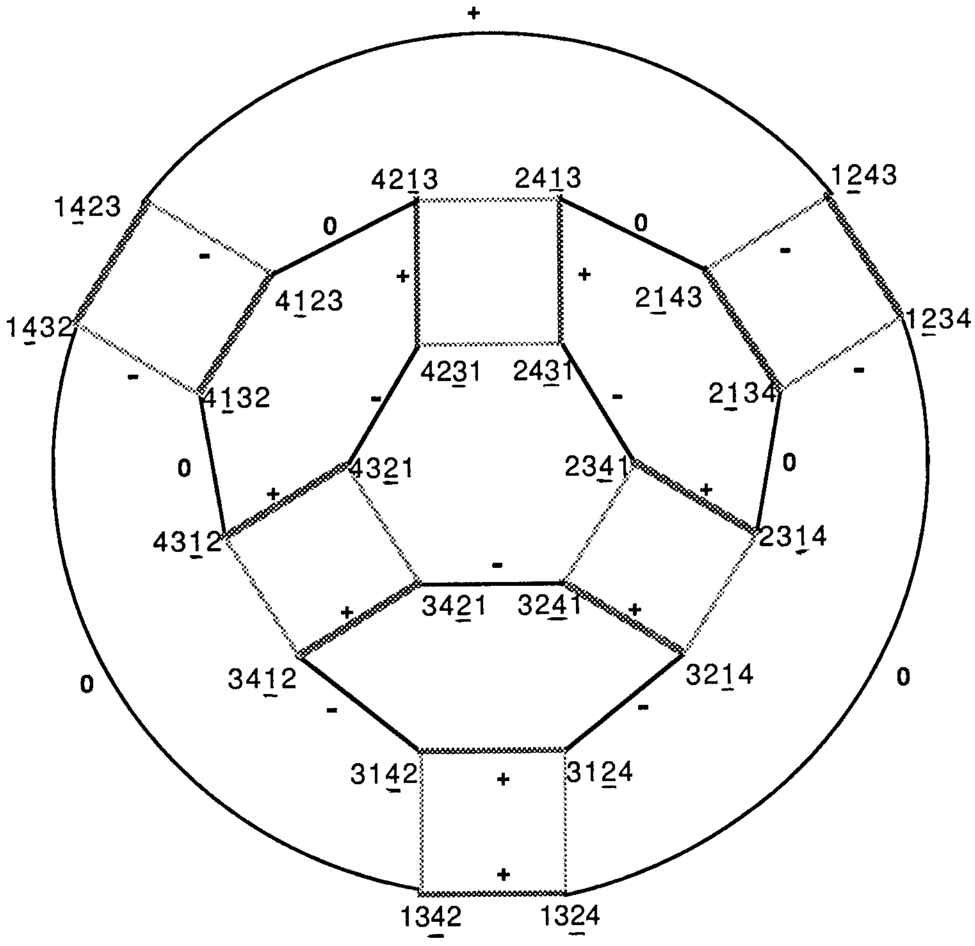
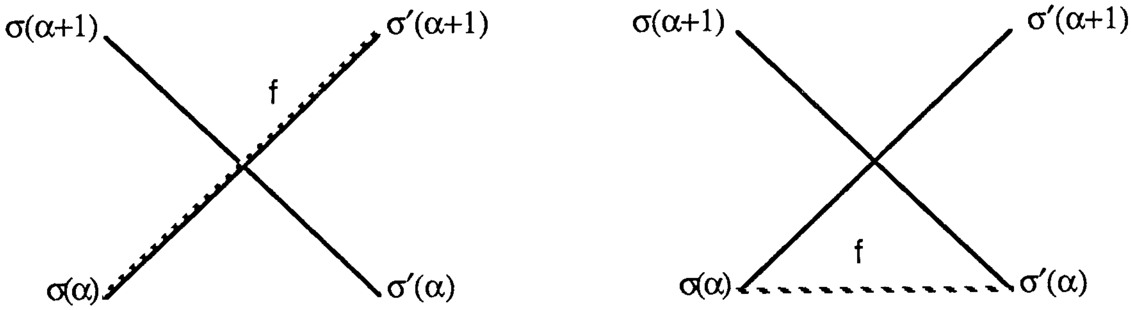


Figure 2: The cpl map associated with $x_1x_2 + x_2x_3x_4 + x_1x_4$

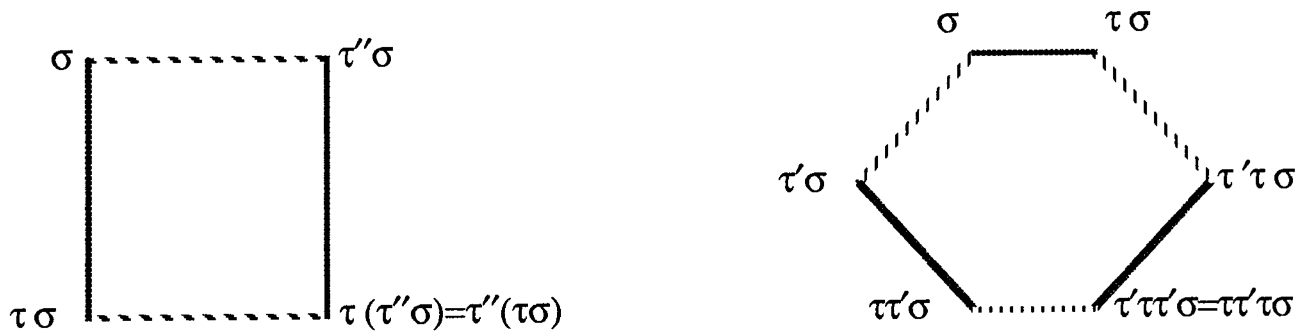
Figure 3: The classification of singular edges



A zero singular edge for f

A positive singular edge for f

Figure 5: The form of the two basic cosets



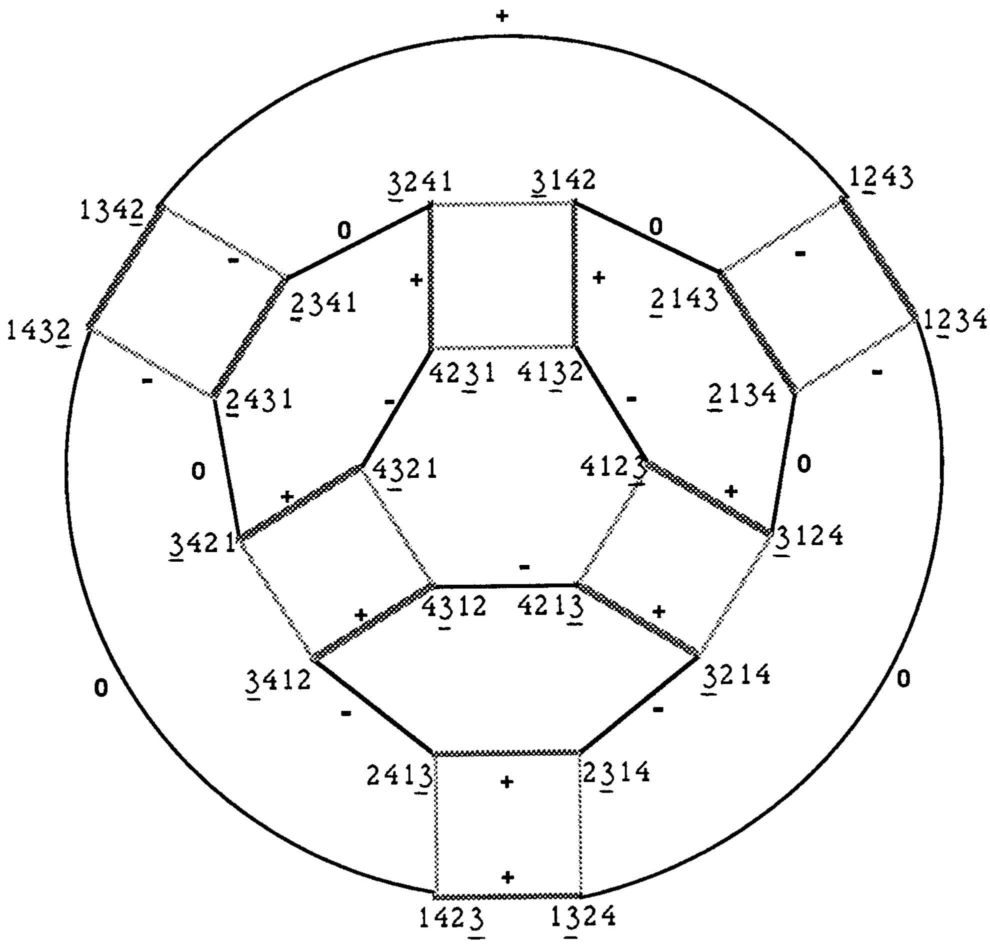


Figure 4: The dual cpl map associated with $x_1 x_2 + x_2 x_3 x_4 + x_1 x_4$

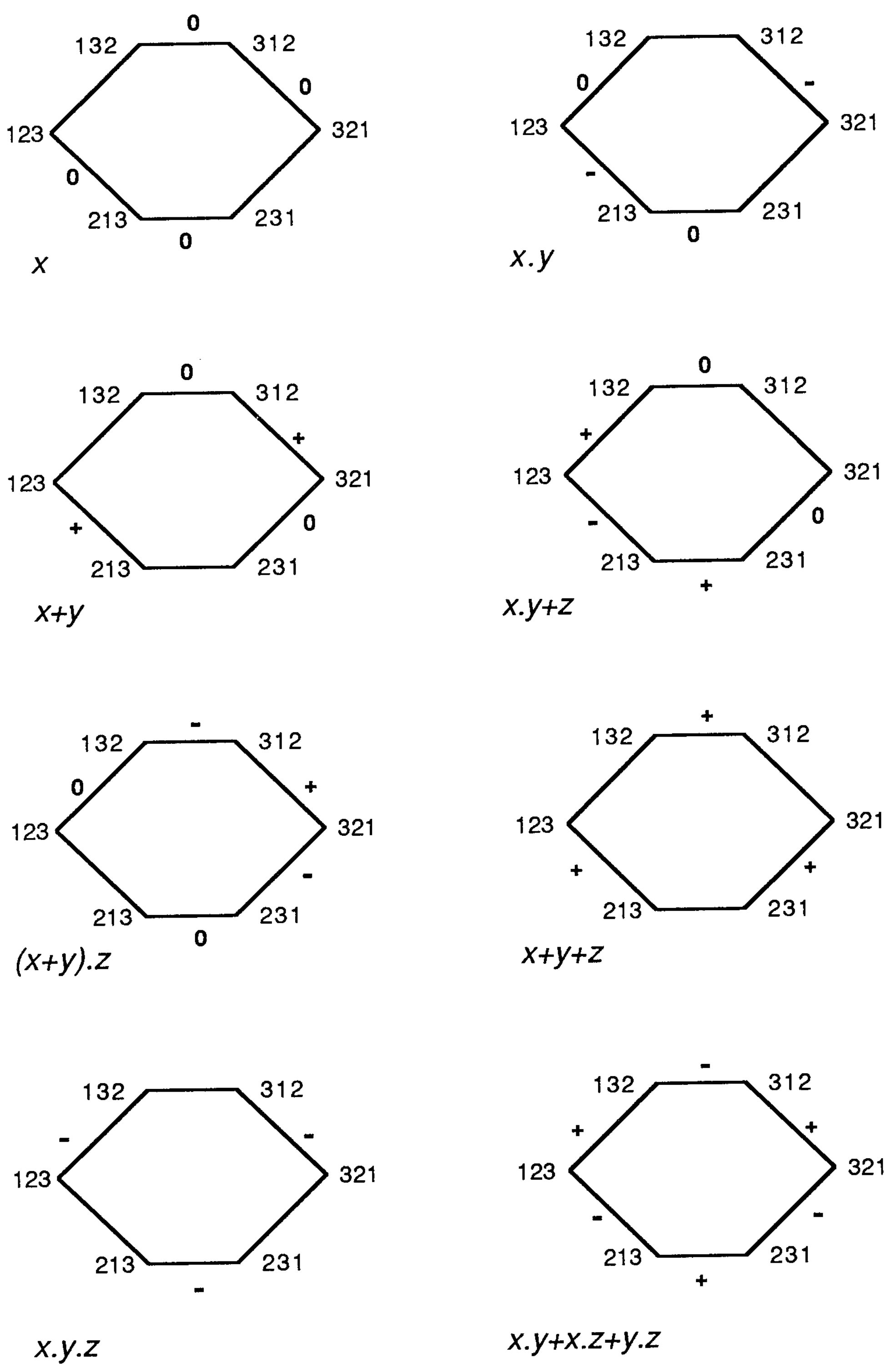


Figure 6: Patterns of singular edges for functions in FDL(3)

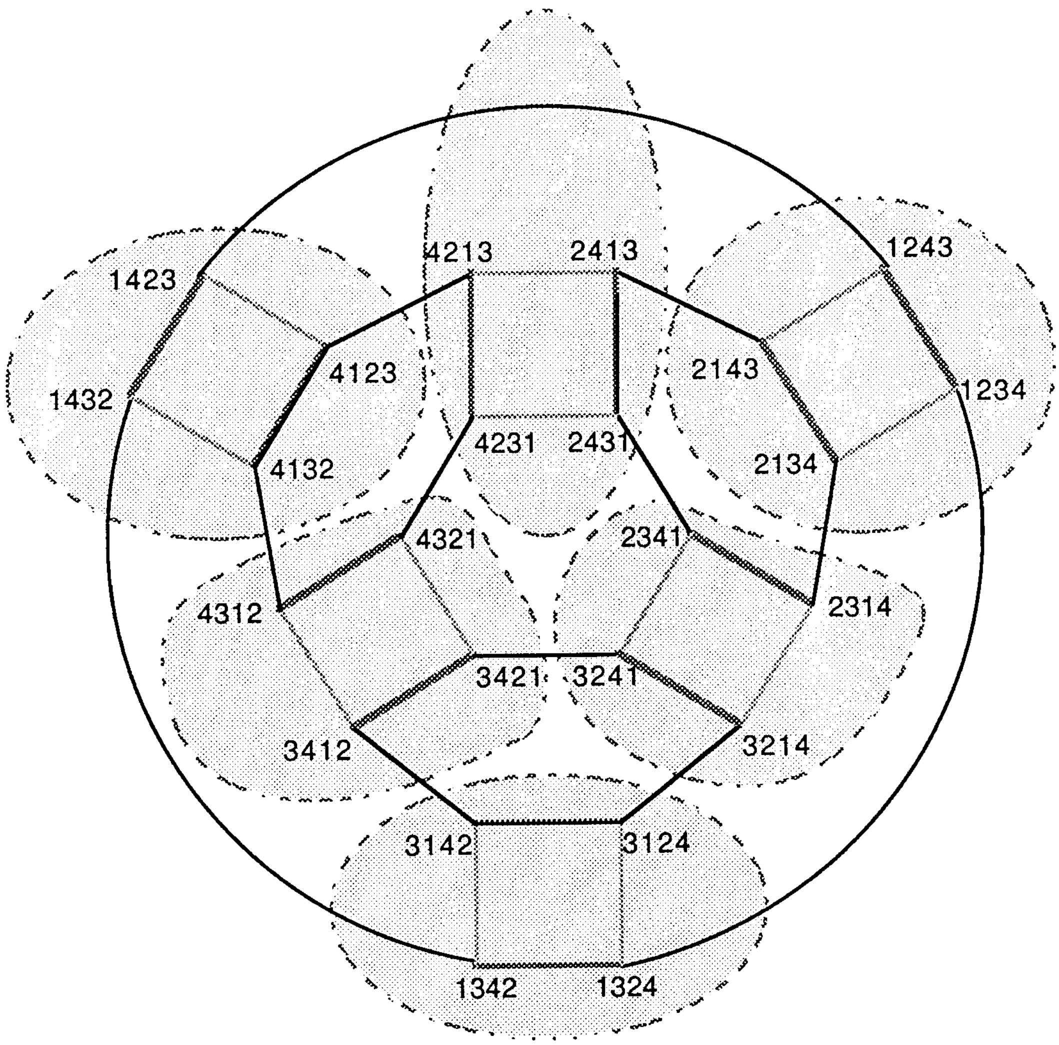


Figure 7a: The regions defined by the prime implicants of threshold 2 in FDL(4)

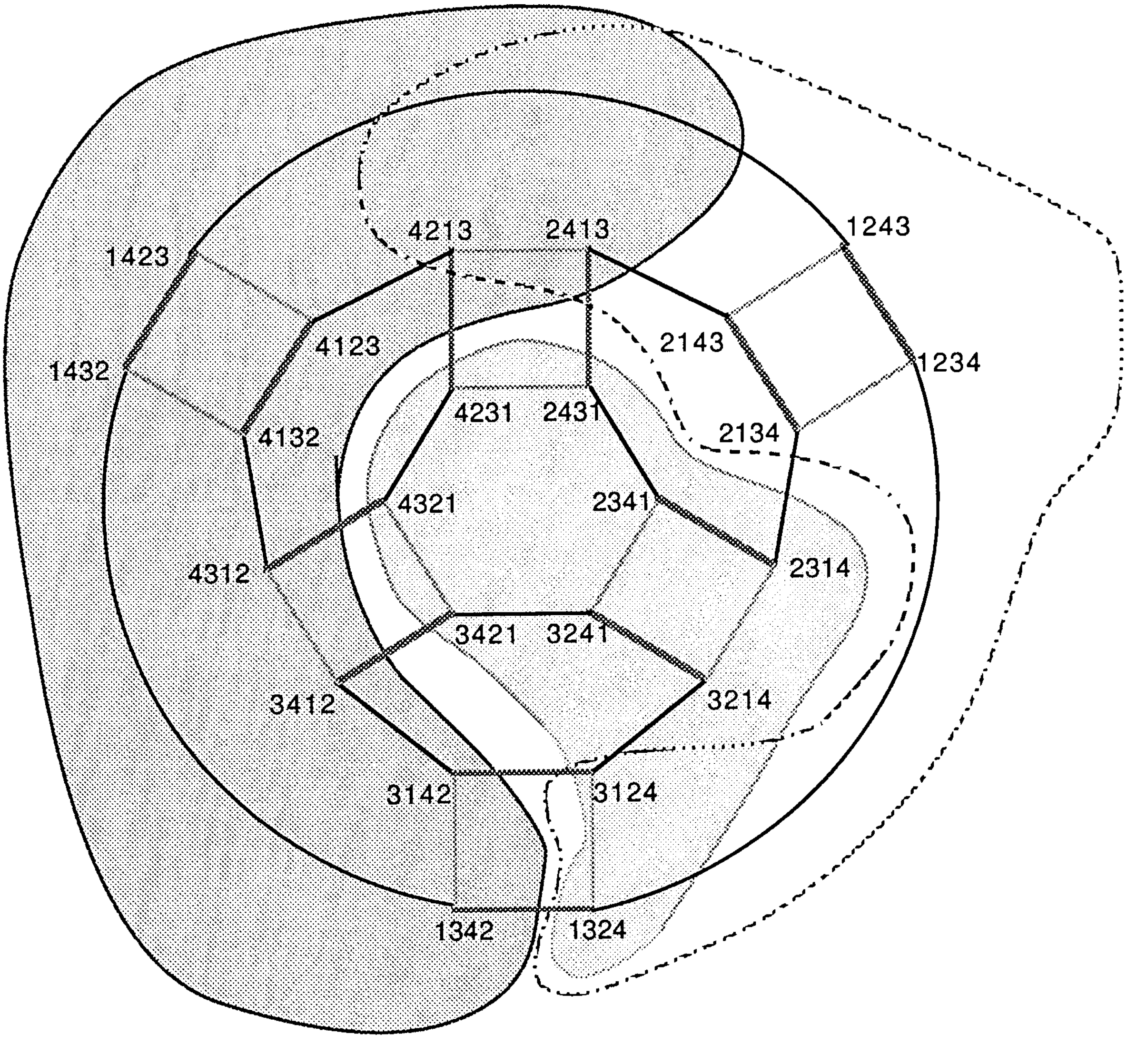


Figure 7b: The regions defined by the prime implicants of the function $x_1x_2 + x_2x_3 + x_1x_4$

Figure 8: The dual singular cycles of length 18 and alternation index 6

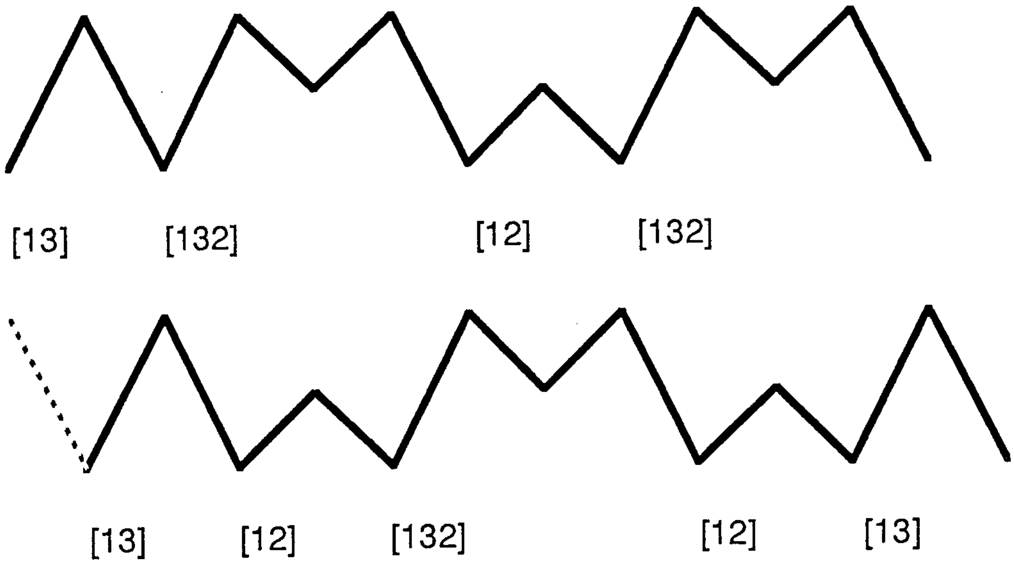
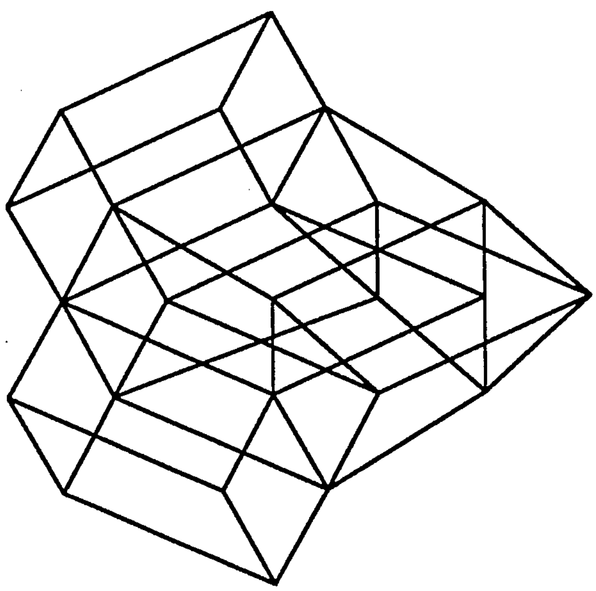


Figure 11: A lattice interval in FDL(5) and its associated configuration



	3	1	1
	2	1	1
23	*	5	4
	45		

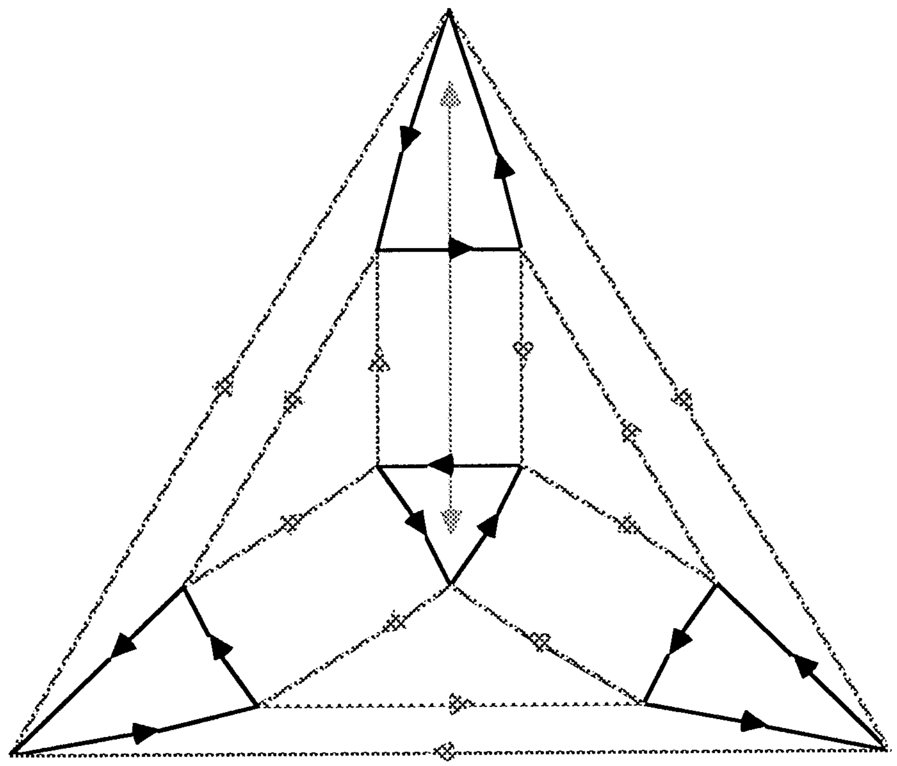


Figure 9: The alternating group as generated by [12] and [13]

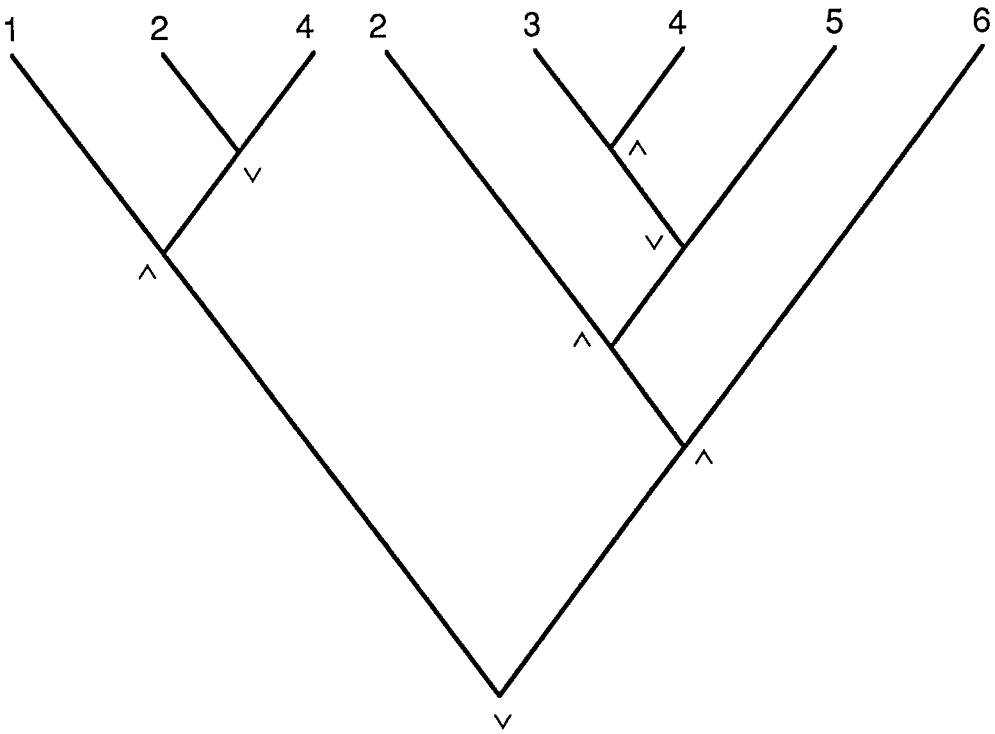
Figure 10a: A boolean formula, and an associated partition

14	4	1	1	1
4	24	2	3	6
1	2	12	1	1
5	2	2	5	6

Generalised rectangles

1	1	14	1	4
1	1	1	12	2
6	3	4	2	24
6	5	5	2	2

After rearrangement



$(1 \wedge (2 \vee 4)) \vee (2 \wedge 6 \wedge (5 \vee (3 \wedge 4)))$

Figure 10b: A simple circuit, and the associated labelling

1	2	12
3	23	2
13	3	1

1acdef	2acef	12acdef
3bcdef	23bcef	2bcdf
13def	3ef	1df

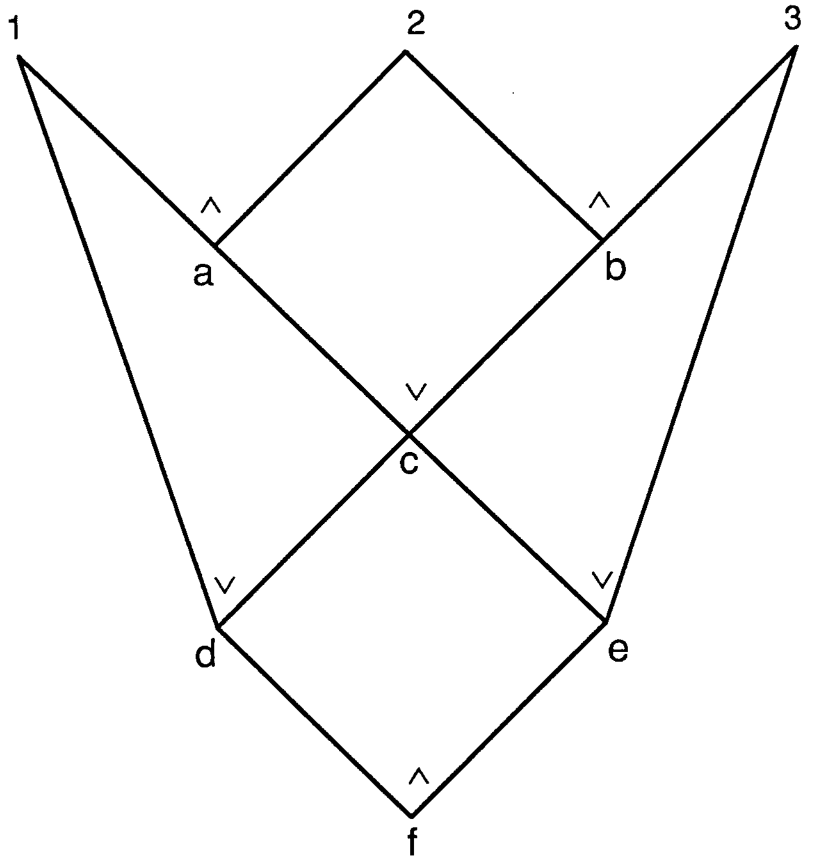


Figure 12: Two nested singular cycles

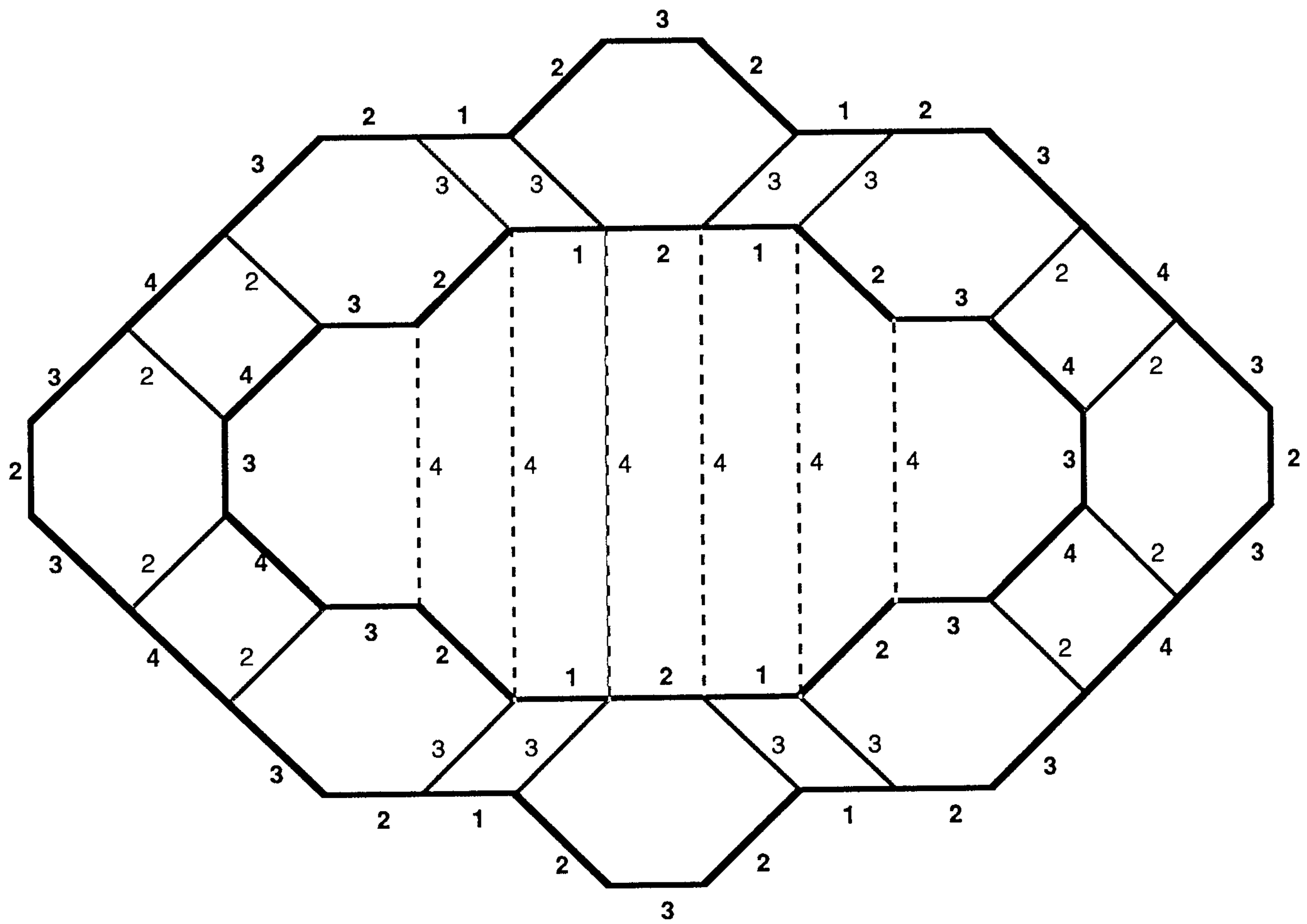


Figure 13: The effect of input substitution on singular cycles

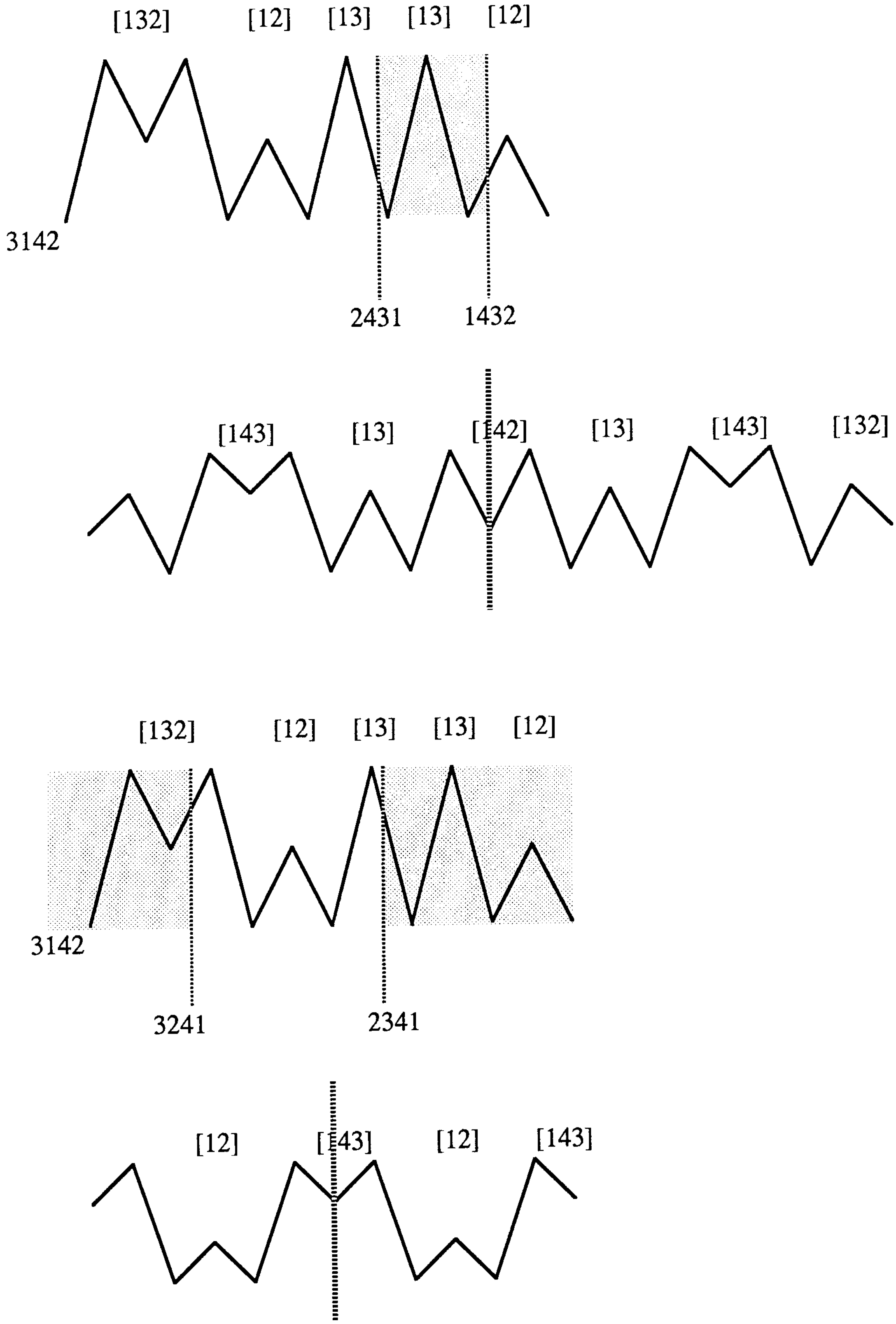
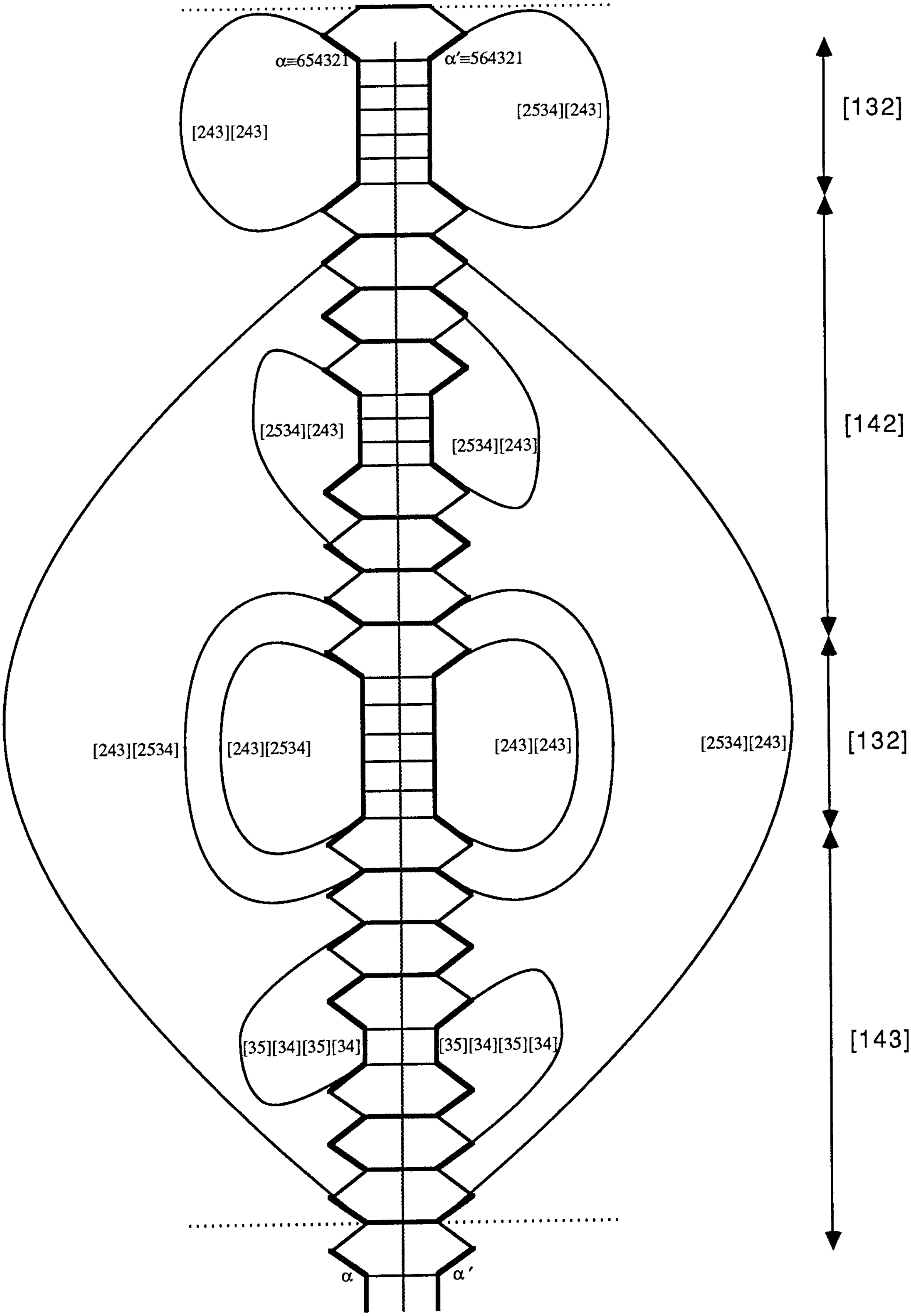


Figure 14: Synthesising a cycle by identification of inputs



Appendix

Some examples of relations arising as singular cycles:

6_3	$[12]^3$		1	2	*		
			3	*	2		
			*	3	1		
12_4	$[132]^2$	2	3	4	*	*	
	$([12][13])^2$		1	1	2	2	
		1	*	*	3	4	
18_5	$[132][1423]$		2	2	1	1	
			2	*	4	3	
		1	5	4	*	*	
		1	5	3	*	*	
					5	5	
18_6	$[13][132][12][132]$		*	4	2	2	
	$[13][12][132][12][13]$		4	*	1	3	
			2	1	*	2	
20_6	$([143][12])^2$		3	1	1		
			2	1	1		
		23	*	5	4		
			45				
24_6	$[1423]^2$	2	*	*	5	4	
	$([14][132])^2$	2	*	*	5	3	
			2	2	1	1	
		1	5	4	*	*	
		1	5	3	*	*	
24_8	$([12][13][132])^2$		1	2	2		
			3	*	2		
			3	3	4		
24_9	$([12][132])^3$		1	2	2	*	
			3	2	*	2	
			4	*	2	2	
			*	4	3	1	
28_6	$([142][13])^2$		4	4	1	1	
			2	2	1	1	
		2	*	*	5	3	
		4	*	*	5	3	
			3	5			
28_6	$[143][15243]$	3	5	4	2	*	*
	$[1523][1423]$		1	1	2	2	2
		1	*	*	6	3	3
		1	*	*	6	4	5
			6	6			

30_8	[142][132][143][132]		2	2	2	1	1	
	[1423][12][1423][13]		3	*	4	1	*	
			5	4	*	*	1	
		2	*	*	*	5	3	
				3	5			
32_8	$([132][154])^2$	134	5	6	*	*		
	$([12][1534])^2$		2	2	1	1		
			2	2	3	4		
					56	56		
32_8	[143][132][143][142]		5	3	*	*	1	
	[13][14][12][1423][12][14]		2	2	1	1	1	
		2	*	*	3	5	4	
34_7	[15243][1534]	2	4	5	6	*	*	
	[15][1423][15][132]		3	3	6	2	*	
			1	1	1	2	2	
		13	*	*	*	5	6	
		13	*	*	*	4	6	
36_8	$([132][14]^2)^2$		5	5	2	*		
	$([14][1423])^2$		1	1	*	2		
			*	*	1	5		
			3	4	2	2		
36_9	[12][14][132][14][13][1423]		*	1	2	2		
	[143][132][142][1423]		*	3	*	2		
			3	*	3	5		
			1	*	4	5		
			2	5				
40_8	$[15243]^2$	2	4	5	6	*	*	*
	$([15][1423])^2$	2	3	3	6	*	*	*
			1	1	1	2	2	2
		1	*	*	*	3	3	6
		1	*	*	*	4	5	6
40_{10}	[132][143][13][142][13][143]		5	5	*	2	2	
	[12][142][13][142][12][1423]		1	1	2	2	2	
			4	3	2	*	*	
			*	*	5	4	3	
42_{12}	$([12][1423])^3$		4	2	*	*	2	
	$([143][132])^3$		3	*	2	*	2	
			1	*	*	2	2	
			*	3	4	*	5	
			*	*	1	3	5	
			*	1	*	4	5	
			5					
44_8	$([143][152])^2$	1	*	*	6	3	*	*
	$([1523][14])^2$		1	1	2	2	1	1
		6	4	5	*	2	*	*
		3	*	*	2	*	4	5
			3	3			6	6

44_{10}	$([1645][132])^2$	*234*	7	5	*	*		
		234	7	6	*	*		
			1	1	2	2		
			1	1	3	4		
					567	*567*		
48_{10}	$([1423][165])^2$	*1345*	6	7	*	*		
	$([16354][12])^2$		2	2	1	1		
			2	2	5	3		
			2	2	5	4		
					67	*67*		
48_{12}	$([1423][12][14][13])^2$		*	1	4			
	$([142][143][132])^2$		1	1	2			
			3	5	5			
			4	5	*			
48_{13}	[143][132][143][1423][143][132]		1	1	1	2	2	
	[1423][12][14][132][14][12][1423][12]		*	*	1	4	3	
			*	1	*	5	3	
		2	3	4	5	*	*	
52_{12}	[14][12][142][13][142][12][14][1423]		2	2	1			
	[142][13][143][14][132][14][143][13]		2	*	5			
			3	5	5			
			4	5	5			
52_{14}	$([143][132][143][13])^2$		*	1	1	1	2	2
	$([12][142][12][1423])^2$		1	*	1	1	4	4
			1	1	*	*	5	3
			2	4	3	5	*	*
54_{13}	[13][143][14][132][14][142][12][1423]		5	1	4	*		
	[14][12][142][132][143][13][14][1423]		2	2	*	4		
			2	*	2	1		
			5	1	3	1		
56_{10}	[152][143][152][15243]		1	1	2	2	2	
	[14][15][1423][15][14][1523]	*1*	*	*	3	3	6	
		2	6	3	*	*	*	
			*	1	5	4	6	
60_{10}	$([162534])^2$	*1*	4	6	7	*	*	*
	$([16][15243])^2$	*1*	5	6	7	*	*	*
		1	3	3	7	*	*	*
			2	2	2	1	1	1
		2	*	*	*	7	3	3
		2	*	*	*	7	6	5
		2	*	*	*	7	6	4
60_{10}	$([1635][142])^2$	*23*	*	*	7	4	*	*
		23	*	*	7	6	*	*
			2	2	1	1	2	2
			3	5	1	1	5	3
		25	*	*	7	6	*	*
		25	*	*	7	4	*	*
			47	*47*			*67*	*67*

60_{12} [16354][132][16254][132]
[1645][1523][1645][1423]

	*	*	1	4	3
	1	1	1	2	2
2	4	4	7	*	*
23	5	6	7	*	*
				567	*567*

72_{15} [1523][142][1534][132][1524][142]
[1523][142][1534][132][1524][143]

	1	1	2	2
	1	4	4	5
2	6	4	*	*
2	6	3	*	*
			6	*6*

72_{15} $([13][15243])^3$
$([153][1423])^3$

	2	*	2	2	1
	2	2	*	2	4
	2	2	2	*	5
1	6	*	3	3	*
4	6	3	*	3	*
5	6	3	3	*	*
	6	1	4	5	*
					6

88_{17} [153][142][153][132][152][143][152][1423]
[15243][14][1523][14][1534][13][1524][13]

2	4	6	*	*	*
	1	1	2	2	2
	1	1	3	5	5
	*	1	4	5	*
				6	*4*

96_{18} [132][15][14][1534][13][15][142][15][13][1534][14][15]
[1534][152][132][153][1524][153][132][152]

*	*	6	1
6	6	6	5
4	3	2	2
1	1	2	*

100_{20} [154][142][1534][132][153][1523][1534][1524]
[12][15][142][15][132][15][143][15][13][1534][132][1524]

	6	3	*	1	*
	6	5	1	*	*
	2	2	1	1	1
2	*	*	4	5	6
2	*	*	3	5	6

114_{20} [15243][14][154][15][142][15][1534][15][132][15][153][13]
[153][13][15][1534][15][132][15][1524][15][12][152][1423]

1	1	*	4	1
1	1	1	2	*
6	5	4	*	2
6	3	3	2	2

132_{22} [16243][152][1645][142][16253][142][1645][152]
[1625][132][1635][1524][1635][132][1625][1534]

	*	*	1	6	3
	1	1	1	2	2
2	3	3	7	*	*
2	4	5	7	*	*
				7	*7*

144_{21} $([14][1726354])^3$
$([174][162534])^3$

	2	2	*	2	2	1
	2	2	2	*	2	5
	2	2	2	2	*	6
1	8	3	*	3	3	*
5	8	3	3	*	3	*
6	8	3	3	3	*	*
1	8	7	*	4	4	*
5	8	7	4	*	4	*
6	8	7	4	4	*	*
	8	7	1	5	6	*
						78