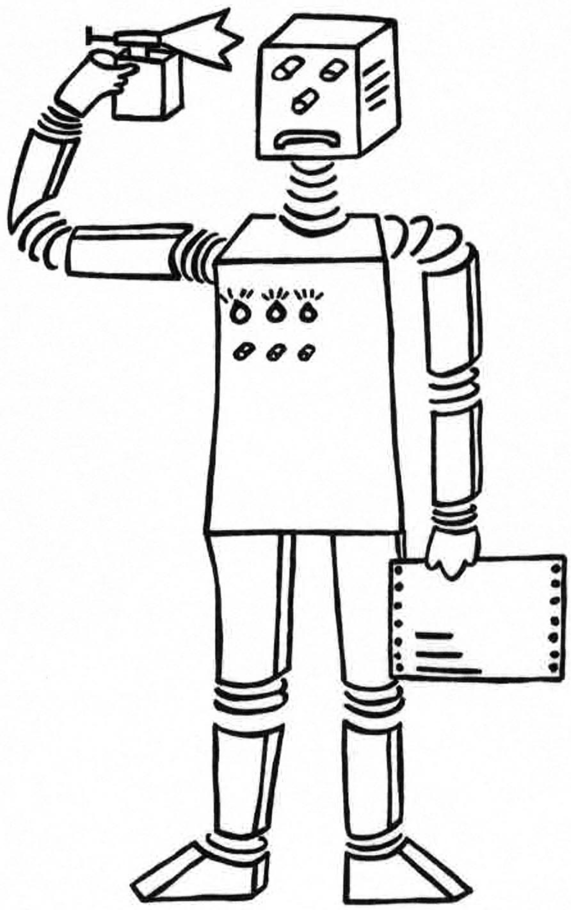
SEKI-REPORT

Artificial
Intelligence
Laboratories

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



An Optimized Transformation into Conjunctive (or Disjunctive) Normal Form

Rolf Socher

SEKI-Report SR-87-13 December 1987

An Optimized Transformation into Conjunctive (or Disjunctive) Normal Form

Rolf Socher

SEKI-Report SR-87-13 December 1987

An Optimized Transformation into Conjunctive (or Disjunctive) Normal Form

Rolf Socher, Fachbereich Informatik, Universität Kaiserslautern
Postfach 3049, D-6750 Kaiserslautern, W.-Germany
net-address: UUCP ..!mcvax!unido!uklirb!socher

Abstract:
Resolution based theorem proving systems require the conversion of predicate logic formulae into clausal normal form. One step of all procedures performing this transformation is the multiplication into conjunctive normal form. In general this is a critical step, since it can result in an exponential increase in the size of the original formula. In general the resulting clausal normal form even contains many redundant clauses. This paper presents a multiplication algorithm that avoids the generation of these redundant clauses. It is shown that the set of clauses generated by this algorithm is the set of prime implicants (in the sense of Quine) of the original formula. Especially in those cases where the usual multiplication algorithm produces a contradictory set of ground clauses the improved algorithm generates the empty clause.

1. Introduction

Most resolution based theorem proving systems require that the logical formulae, which are to be proved, should be converted into clausal normal form. This transformation usually takes several steps including the elimination of implications and equivalences, skolemization and sometimes the resulting formula can be splitted into several easier to prove subformulae [EW83]. In any case, the last step of the procedure consists in the multiplication of formulae containing only the connectives $\wedge$, $\vee$ and $\neg$ to disjunctive normal form or to conjunctive normal form, respectively. Usually the transformation into disjunctive form is required for formulae to be tested for splitting whereas conversion into conjunctive form is necessary for the single splitparts. In general this multiplication is the most critical step of the algorithm, as it can result in an inflation of the original formula. For a multiplication of a disjunctive form $\mathcal{D}$ to a conjunctive form $\mathcal{C}$ (or vice versa) the number of subformulae of $\mathcal{C}$ depends exponentially on the number of subformulae of $\mathcal{D}$. But in general most of the resulting formulae are redundant, as the following example shows:

1.1 Example:

The propositional formula

$$\mathcal{F} = (p\wedge r\wedge s) \vee (p\wedge q\wedge s) \vee (q\wedge r\wedge s)$$

is to be transformed into CNF.

In the following we drop the $\vee$ and write conjunctions as sets.

Multiplying yields

(1) {ppq, ppr, pps, pqq, pqr, pqs, psq, psr, pss,
rpq, rpr, rps, rqq, rqr, rqs, rsq, rsr, rss,
spq, spr, sps, sqq, sqr, sqs, ssq, ssr, sss}

We call this form the totally multiplied form of $\mathcal{F}$.

Using the commutativity and idempotence of $\vee$ this clause set can be simplified to

(2) {pq, pr, ps, pq, pqr, pqs, psq, psr, ps,
rpq, rp, rps, rq, rq, rqs, rsq, rs, rs,
spq, spr, sp, sq, sqr, sq, sq, rq, s}

In this formula all multiple occurrences of clauses and all clauses that are subsumed by some other clause, are redundant. (A clause C is said to subsume another clause D, if C is a proper subset of D). Deleting these redundant clauses yields

(3) {pq, pr, rq, s}

In this particular example the formula $\mathcal{F}$ to be transformed into CNF is given in DNF. Since this is the basic case for our algorithm, we first concern with the transformation from DNF into CNF. The transformation from CNF into DNF is symmetric and therefore need not be considered.

The example shows that the multiplication process produces many terms that can be deleted by subsequent simplification steps. Would it not be better to avoid the generation of these redundant terms in the first place? Ideally the output of such an algorithm should be a minimal representation of the original formula. In this paper we present an algorithm that multiplies formulae into clausal normal form without producing redundant clauses. The output $\mathcal{P}$ of the algorithm, given a formula $\mathcal{F}$, is minimal in the following sense: $\mathcal{P}$ is the set of all clauses implied by $\mathcal{F}$, such that for each clause C in $\mathcal{P}$ there is no subset of C that is implied by $\mathcal{F}$. Thus $\mathcal{P}$ is logically equivalent to $\mathcal{F}$. Such a set is called a set of *prime implicants* of $\mathcal{F}$ and this set is uniquely determined by $\mathcal{F}$. Especially if $\mathcal{F}$ is an unsatisfiable ground formula then $\mathcal{P}$ consists only of the empty clause. The standard method to obtain the set of prime implicants from a clause set $\mathcal{C}$ is the successive computation of resolvents and the deletion of subsumed clauses [Qu59].

In general the set of prime implicants is not absolutely minimal, as the following example shows: Let $\mathcal{F}$ be the clause set $\{C_1, C_2, C_3, C_4\}$, where $C_1=\{p, q, r\}$, $C_2=\{-q, r\}$, $C_3=\{q, s\}$, $C_4=\{r, s\}$. The clause $D=\{p, r\}$ is a subset of C_1 and is implied by $\mathcal{F}$. Thus the set $\{D, C_2, C_3, C_4\}$ is the set of prime implicants of $\mathcal{F}$, but it is not minimal, since the set $\{D, C_2, C_3\}$ is already equivalent to $\mathcal{F}$. Such a minimal subset of the set of prime implicants is called a simplest equivalent. In general this minimal representation of a formula is not unique. However, if the set $\mathcal{P}$ of prime implicants of a formula $\mathcal{F}$ does not contain a pair of resolvable clauses, then $\mathcal{P}$ is already a simplest equivalent of $\mathcal{F}$.

This paper is concerned with the simplification of formulae on the basis of the equality of some elements of a formula, where the equivalence relation itself is deliberately left unspecified. We formulate our results in the language of pure propositional logic, but the reader should be aware that the propositional variables can stand for several things:

1.2 Example:

The simplification of $p\wedge(p\vee q)$ to p means that any formula, which is a conjunction between an element p and a disjunction having an element equal to p as disjunct, is reduced to p.

a) If the elements p and q are interpreted as atomic formulae of predicate logic together with syntactic equality, then the formula $Px\wedge(Px\vee Qa)$ can be reduced to Px under this particular interpretation, whereas the formula $Px\wedge(Py\vee Qa)$ cannot.

b) Suppose that the propositional variables are interpreted as formulae and the equivalence relation is taken to be equality up to renaming of bound variables. Then $Px\wedge(Py\vee Qa)$ is equivalent to $(\forall xPx)\wedge((\forall yPy)\vee Qa)$ and the formulae $\forall xPx$ and $\forall yPy$ are equal up to renaming of bound variables, hence the reduction to $\forall xPx$ is possible.

Using example 1.1 we now describe how this algorithm works:

We write the formula $\mathcal{F}$ as a 3×4-matrix M and label the rows of M with the variables of $\mathcal{F}$ and the columns of M with the numbers of the subformulae of $\mathcal{F}$. We set M(p,k)=1 if the k-th term of $\mathcal{F}$ contains the propositional variable p and M(p,k)=0 otherwise. This results in the following matrix for $\mathcal{F}$

M =

	1	2	3
p	1	1	0
q	0	1	1
r	1	0	1
s	1	1	1

The conjunctions of the original formula $\mathcal{F}$ correspond to the columns of the matrix. The clauses of the totally multiplied form of $\mathcal{F}$ correspond to the paths through the matrix. A path is obtained by taking in each column of M a nonzero entry and writing down the variable of the corresponding row. Thus a path is a sequence (p_1, p_2, p_3) where $M(p_1,1) = M(p_2,2) = M(p_3,3) = 1$. For instance (p,p,q), (p,s,r) or (s,s,s) are paths through M. We say that a path P subsumes another path Q, if the set of variables of P is a subset of the set of variables of Q, i.e. if the clause corresponding to P subsumes the clause corresponding to Q.

The terms of the reduced form (2) of $\mathcal{F}$ can be obtained as follows: The paths are computed just as before but only the first occurrence of a variable in the path is counted: Then (p,q) is the path (p,p,q) or (r,q) is the path (r,q,r) in M.

The terms of the totally reduced form (3) correspond to the following subset of paths of M: If a path P of M contains two entries of the same variable in two different columns i and j, for instance (p,p,q), then all paths differing from P only in either i or j, namely the paths (p,p,q), (p,s,q), (r,p,q) and (s,p,q), are subsumed by (p,q) or equal to (p,q) up to permutation. The generation of these redundant paths can be avoided in the following way: Having developed a partial path $(p_1,p_2,...,p_i)$ as before take a 1-entry in the column i+1 only if $M(p_j, i+1)=0$ for all j with $1 \le j \le i$, otherwise continue with column i+2.

If we develop the paths beginning with the row s in our example, we see that all columns except the first can be disregarded.

Once we developed the path containing only the variable s, we can cancel the whole row: all paths starting with a different variable and containing s are subsumed by the path (s).

This is the analogon to the transformation:

$(p \wedge r \wedge s) \vee (p \wedge q \wedge s) \vee (q \wedge r \wedge s) \rightarrow s \wedge ((p \wedge r) \vee (p \wedge q) \vee (q \wedge r))$.

Developing first the paths starting from s (i.e. the paths (s,...)) avoids computing unnecessary paths: Beginning with the paths starting from p, one obtains the paths (p,q), (p,r), (p,s). But (p,s) is redundant, since it is subsumed by (s). Later we will prove that the strategy of developing first a row with a maximal number of 1-entries always produces a minimal set of paths.

The computation of $\mathcal{P}$ from the matrix M is done as follows:

1. At the beginning the result set $\mathcal{P}$ is empty.

   First we develop the s-row. The second and third column don't have to be considered, since they have a 1-entry at s. Thus we obtain the path (s), add it to $\mathcal{P}$ and cancel the s-row from the matrix. Now we have $\mathcal{P}=\{(s)\}$ and

M =

	1	2	3
p	1	1	0
q	0	1	1
r	1	0	1

2. Next we develop the p-row obtaining the two paths (p,q) and (p,r) (the second column can be canceled, since it contains a 1-entry for p), add them to the solution set and obtain $\mathcal{P}=\{(s),(p,q),(p,r)\}$. Now the p-row can be canceled also and the remaining matrix is

M =

	1	2	3
q	0	1	1
r	1	0	1

   This matrix corresponds to the formula

   $$\mathcal{F} = r \vee q \vee (q \wedge r)$$

   Now the absorption law is applicable to $\mathcal{F}$ yielding

   $$\mathcal{F} = r \vee q$$

   The analogon in our procedure is the deletion of the third column of M and hence

M =

	1	2
q	0	1
r	1	0

3. Now the only remaining path in M is (r,q). This path is added to the solution set and the result is

$\mathcal{P} = \{ (s), (p,q), (p,r), (r,q)\}$.

This is the set of prime implicants of the original formula $\mathcal{F}$ and since no pair of clauses is resolvable, it is already a simplest equivalent of $\mathcal{F}$.

2. Normal Form Matrices

We use the following notions of the propositional calculus:

$\mathbb{P}$ is a set of propositional variable symbols. $\mathbb{L}$ is the set of all propositional literals (+p and -p).

For any object o containing variables we define $\mathbb{P}(o)$ as the set of all **variables occurring in** o. $\mathbb{L}(o)$ is the set of all **signed variables** occuring in o. $\mathbb{P}^+(o)$ is the set of all **variables** occurring with **positive sign** in o, $\mathbb{P}^-(o)$ is the set of all **variables** occurring with **negative sign** in o and $\mathbb{P}^\pm(o) = \mathbb{P}^+(o) \cap \mathbb{P}^-(o)$.

We write $\mathcal{F} \equiv \mathcal{G}$ to denote that the propositional formulae $\mathcal{F}$ and $\mathcal{G}$ are logically equivalent.

<u>2.1 Definition:</u>

A normal form matrix (NF-matrix) M is an n×k-matrix over the set {0,1}. The rows of M are labeled with different literals from $\mathbb{L}$. We write M(p,i) for the element of M in the p-th row and the i-th column. We define

$F_C(M) := \bigwedge_{i=1..k} \bigvee_{p:M(p,i)=1} p$, the formula in conjunctive normal form belonging to M and

$F_D(M) := \bigvee_{i=1..k} \bigwedge_{p:M(p,i)=1} p$, the formula in disjunctive normal form belonging to M.

We call these formulae the **totally multiplied forms** of M.

<u>2.2 Definition:</u>

Let M be an NF-matrix.

(i) We say that a column i of M **absorbs** a column j of M, if $M(p,i) \leq M(p,j)$ for all $p \in \mathbb{L}(M)$.

(ii) A column i of M is called **tautological**, if there is a $p \in \mathbb{L}(M)$ with M(p,i)=M(-p,i)=1.

(iii) A **complete path** P through M is a sequence $(p_1,...,p_n)$ of variables such that $M(p_i,i)=1$ for each i and $\mathbb{P}^\pm(P) = \emptyset$.

A **path** through M is a subsequence of a complete path. If P and Q are paths of M, then we denote their concatenation by $P \oplus Q$.

We write P(i) for the i-th element of the path P.

(iv) Let $\mathcal{P}$ be a set of paths through a matrix M.

We define

$F_C(\mathcal{P}) := \bigwedge_{P \in \mathcal{P}} \bigvee_{p \in L(P)} p$, the formula in conjunctive normal form belonging to $\mathcal{P}$ and

$F_D(\mathcal{P}) := \bigvee_{P \in \mathcal{P}} \bigwedge_{p \in L(P)} p$, the formula in disjunctive normal form belonging to $\mathcal{P}$.

In the following M is a NF-matrix.

<u>2.3 Lemma:</u>

The set $\mathcal{P}$ of all complete paths through a NF-matrix M represents the totally multiplied form of the formula belonging to M, i.e.

$F_C(\mathcal{P}) \equiv F_D(M)$ and $F_D(\mathcal{P}) \equiv F_C(M)$ ■

2.4 Definition:

We say that a path P **subsumes** a path Q, if $\mathbb{L}(P) \subset \mathbb{L}(Q)$. We write P<Q if P subsumes Q and P≤Q if P=Q or P<Q.

2.5 Remark:

The relation < is a partial order on the set of all paths through M.

The next lemma says that subsumed paths can be canceled from path sets and absorbed columns can be canceled from NF-matrices:

2.6 Lemma:

(i) Let $\mathcal{P}$ be a set of paths through M and P,Q∈ $\mathcal{P}$ with P < Q. Then
$F_C(\mathcal{P}) \equiv F_C(\mathcal{P} - \{Q\})$

(ii) Let i and j be columns of M such that i absorbs j. Let M' be the NF-matrix obtained by canceling the column j of M. Then
$F_C(M) \equiv F_C(M')$ ■

2.7 Lemma:

(i) Let p be a row of M such that M(p,1)=1. Each complete path P containing p at a position k≠1with p≠P(1) is a subsumed path.

(ii) Let P be a complete path of M with P(j)=p. If there is a column i≠j, such that M(p,i)=1 and p≠P(i), then P is a subsumed path.

Proof:

(i) Since M(p,1)=1, the path Q defined by Q(1)=p and Q(j)=P(j) for j>1 is a complete path of M. Since $p \in \mathbb{L}(P)$, $\mathbb{L}(Q) = \mathbb{L}(P) \setminus \{P(1)\} \subset \mathbb{L}(P)$.

(ii) Since M(p,i)=1, the path Q defined by Q(i)=p and Q(j)=P(j) for j≠i is a complete path of M. Since $p \in \mathbb{L}(P)$, $\mathbb{L}(Q) = \mathbb{L}(P) \setminus \{P(i)\} \subset \mathbb{L}(P)$. ■

3. The Multiplication Algorithm

We are now ready to formulate the algorithm that performs an optimized multiplication between conjunctive and disjunctive normal form.

Algorithm

Transform

Input: An NF-matrix M .

Output: A set $\mathcal{P}$ of paths through M such that $F_C(M) \equiv F_D(\mathcal{P})$ and $F_D(M) \equiv F_C(\mathcal{P})$

1. $\mathcal{P} := \emptyset$.

Cancel all tautological columns of M.

2. Cancel all absorbed columns of M.
   If M is now a matrix with only zero entries, go to 5.
3. Take a row p of M that has a maximal number of 1-entries and permute the columns of M in such a way, that M(p,1) = 1. Generate all paths of M with initial part (p) at column 2 and add them to $\mathcal{P}$.
4. Cancel the row p of M and go to 2.
5. Return $\mathcal{P}$.

Generate all paths of **M** with initial part **Q** at column **i**

Input: A matrix M corresponding to a formula $\mathcal{F}$ in conjunctive normal form, a path Q, developed from column 1 to column i-1, and a column i of M.

Output: $\mathcal{Q} = \{P \in \mathcal{P} \mid P$ is output of transform and P has initial part Q$\}$

Remark: All parameters are value parameters, especially the matrix M is unchanged when the procedure has terminated.

1. If i is greater than the last column of M then return {Q}.
2. If there is a $p \in \mathbb{P}(Q)$ such that M(p,i) = 1then i:=i+1; go to 1.
3. $\mathcal{Q} := \emptyset$.
   For all $q \in \mathbb{L}(M) \setminus \{-p \mid p \in Q\}$ such that M(q,i) = 1 do
   3.1 Generate all paths of M with initial part $Q \oplus (q)$ at column i+1;
   3.2 Add these paths to $\mathcal{Q}$.
   3.3 Cancel the row q from M.
4. Return $\mathcal{Q}$.

Since the algorithm may still produce subsumed paths, these are removed after the last step of the transform algorithm. In the following $\mathcal{P}(M)$ denotes the set of paths that can be obtained from the matrix M using the algorithm above and which does not contain subsumed paths.

3.1 Lemma:

Let P,Q be paths generated by the algorithm. If P is a permutation of Q, i.e. if $\mathbb{L}(P) = \mathbb{L}(Q)$, then P=Q.

Proof:

Suppose P≠Q and j is the first index for which P(j)≠Q(j). Since Q is a permutation of P, there is a i≠j with Q(j)=P(i). Suppose i<j. Since j is the first index, such that P(j)≠Q(j) we have Q(j)=P(i)=Q(i), but according to step 2 of the generate algorithm, Q cannot have multiple occurrences of a literal, which is a contradiction. Hence we have i>j. In the same way we get P(j)=Q(m) for some m>j. Therefore P and Q have the following form:

```
         1        j-1  j         i
    P = (P(1),...,P(j-1),P(j),....,P(i),...)
    Q = (P(1),...,P(j-1),P(i),..,P(j),.....)
         1        j-1  j       m
```

Without loss of generality we may assume that P was developed before Q by the algorithm. After the development of P first all other paths beginning with P(1),...,P(j) must have been developed and thereafter the row P(j) must have been deleted according to step 4 of the transform algorithm or step 3.3

of the generate algorithm. Hence the path Q, which contains P(j), could not have been developed and this is a contradiction. ■

Thus two different paths produced by the algorithm indeed correspond to two different clauses.

3.2 Remark:
Lemmata 2.3 to 2.8 together show, that for each complete path P of a matrix M there is a path $Q \in \mathcal{P}(M)$ with $Q \leq P$.
The last step of the algorithm assures that $P \leq Q$ implies $P=Q$ for arbitrary $P,Q \in \mathcal{P}(M)$.

The next lemma justifies step 2 of the transformation algorithm. It says that by developing at first a row with a maximal number of 1-entries, not more paths are produced than by developing any other row that is not maximal. Example 1.1 showed that the converse is not true in general: if at first the s-row is developed, the result contains fewer paths than by developing at first the p-row.

3.3 Lemma:
Let p and q be rows of an n×k NF-matrix M such that $\Sigma_{i=1..k} M(q,i) < \Sigma_{i=1..k} M(p,i)$. Suppose $\mathcal{P}$ is the set of paths obtained by developing the row p at first and $\mathcal{Q}$ is the set of paths obtained by developing the row q at first. If P is any path in $\mathcal{P}$, then there is a path Q in $\mathcal{Q}$, such that Q is only a permutation of P.

Proof:
Without loss of generality we permute the columns of M in such a way, that we have first the set C_1 of columns i with M(p,i)=M(q,i)=1, then C_2 with M(p,i)=1, M(q,i)=0, then C_3 with M(p,i)=0, M(q,i)=1, and last C_4 with M(p,i)=M(q,i)=0. From $\Sigma_{i=1..k} M(q,i) < \Sigma_{i=1..k} M(p,i)$ follows that $C_2 \neq \emptyset$. Furthermore we assume that absorbed columns in M are already deleted.
Then we can write M in the following form:

M =

	C_1	C_2	C_3	C_4
p	1..1	1..1	0..0	0..0
q	1..1	0..0	1..1	0..0
..	....			

We have to construct to each path P in $\mathcal{P}$ a path Q in $\mathcal{Q}$ that is a permutation of P.
Case I: P has the form (p,q,....), where p is from C_1 and q is from C_3.
It is easy to see that the following path Q is in $\mathcal{Q}$: first take q from C_1 and then p from C_2. The rest of the sequence can be obtained in the same way as by developing p first. Moreover Q is only a permutation of P.
Case II: P has the form $(p,x_1,...,x_m,y_1,...,y_n)$, where p is taken from C_1, the x_i are taken from C_3 and the y_j are taken from C_4. We show that the same path P is in $\mathcal{Q}$.
After the development of the p-row this row has been deleted according to step 3. Let M' be the resulting matrix:

$M' =$

	C_1	C_2	C_3	C_4
p	1..1	1..1	0..0	0..0
x_1			1	
⋮				
x_m			1	
y_1				1
⋮				
y_n				1

There is only one possibility that the path $P=(p,x_1,\dots,x_m,y_1,\dots,y_n)$ is not developed in $\mathcal{Q}$: one column j of the set C_3 or the set C_4, from which one of the x_i or y_j of P has been taken, has been canceled in step 2 of the transform algorithm.

The absorbing column h, against which j has been canceled, must be from C_3 or C_4, since $M(p,j)=0$ and $M(p,i)=1$ for all $i \in C_1 \cup C_2$. Suppose that $j \in C_3$. This implies $M(q,j)=1$ and so we have $M(q,j) \geq M(q,h)$. Furthermore we have $M(u,j) \geq M(u,h)$ for all $u \neq q$, since h absorbs j in M'. Together we have $M(u,j) \geq M(u,h)$ for all $u \in \mathbb{L}(M)$, i.e. h absorbs j already in M and this is a contradiction. Thus j must be from C_4 and there must be some y_s with $M(y_s,j)=1$. By an analogous argument h must be from C_3. There must be an x_r such that $M(x_r,h)=M'(x_r,h)=1$. Since j is absorbed by h, we have $M'(x_r,j)=1$, hence also $M(x_r,j)=1$. Thus we have the following situation:

$M =$

	....h......	j....
⋮		
x_r	1	1
⋮		
y_s		1
⋮		

But in this situation the path $P=(\dots x_r,\dots,y_s,\dots)$ could not have been developed according to step 2 of the generate algorithm. Therefore the path P must be in $\mathcal{Q}$. ■

4. The Simplest Equivalent

In this section we show that the algorithm of chapter 3 produces the set of prime implicants of a formula. Moreover it is shown how the core implicants and the absolutely eliminable implicants of a formula can be determined in terms of NF-matrices. This classification of implicants is necessary to find a simplest equivalent of a formula.

4.1 Definition:

(i) A **prime implicant** of a formula $\mathcal{F}$ in conjunctive normal form is a term that is implied by $\mathcal{F}$ and subsumes no shorter term that is implied by $\mathcal{F}$. The prime implicant of a formula $\mathcal{G}$ in disjunctive normal form is defined dually.

(ii) Let P and Q be paths of a matrix M such that $\mathbb{P}^+(P) \cap \mathbb{P}^-(Q) \neq \emptyset$. Let $r \in \mathbb{P}^+(P) \cap \mathbb{P}^-(Q)$. Then we call P and Q **r-resolvable** and their **r-resolvent** R is defined as

$$R \equiv P\backslash\{r\} \oplus Q\backslash\{-r\}$$

We write $R = P +_r Q$. Furthermore we write $R = P+Q$, if there is a $r \in \mathbb{P}^+(P) \cap \mathbb{P}^-(Q)$ with $R = P+_r Q$.

4.2 Lemma:

Let P,Q and R be paths of a matrix such that $P \leq Q$ and P and R as well as Q and R are r-resolvable. Then $P +_r R \leq Q +_r R$ ■

The main interest in prime implicants lies in the fact that a simplest CNF formula $\mathcal{G}$ equivalent to $\mathcal{F}$ is a conjunction of prime implicants of $\mathcal{F}$ [LS80]. The prime implicants of a given CNF formula $\mathcal{F}$ can be obtained by the method of *iterated consensus* described in [Qu59]: The nontautological resolvents of disjunctions of $\mathcal{F}$ are repeatedly formed and added to $\mathcal{F}$. At the same time subsuming terms are deleted. When no new terms can be added that are not subsumed by existing terms, the set of prime implicants has been obtained. Thus the prime implicants of $\mathcal{F}$ are the minimal (with respect to the subsumption order) elements of the set of all clauses that are implied by $\mathcal{F}$.
We now show that a formula $\mathcal{F}$ obtained by the transform algorithm is the set of all its prime implicants.

4.3 Lemma:

Let P and Q be r-resolvable paths from $\mathcal{P}(M)$ and $R=P+_rQ$ such that R is nontautological.
Then there is an $S \in \mathcal{P}(M)$ with $S \leq R$.

Proof:

W.l.o.g. let $P=(r,u_1,..,u_i,w_1,..,w_j)$ and $Q=(-r,u_1,..,u_i,s_1,..,s_k)$, $i,j,k \geq 0$. Then we have $R=(u_1,..,u_i,w_1,..,w_j,s_1,..,s_k)$. We define $U:=\{u_1,..,u_i\}$, $S:=\{s_1,..,s_k\}$ and $W:=\{w_1,..,w_j\}$.
We have to show that there is a path S in $\mathcal{P}(M)$ such that $\mathbb{L}(S) \subseteq \mathbb{L}(R)= U \cup W \cup S$. Let c be any column of M with $M(r,c)=1$. Then $M(-r,c)=0$, since otherwise the column c would be tautological. Since $Q=(-r,u_i,..,u_i,s_1,..,s_k)$ is a path through M and $M(-r,c)=0$, there must be an $x \in U \cup S$ with $M(x,c)=1$.
Analogously if d is any column of M with with $M(-r,d)=1$, then there must be a $y \in U \cup W$ with $M(y,d)=1$.
The same argument shows that for any column e with $M(r,e)=M(-r,e)=0$ there must be a $z \in U \cup W$ with $M(z,e)=1$ and a $z' \in U \cup S$ with $M(z',e)=1$.
Now we have shown that for each column c of M there exists an $p \in U \cup W \cup S$ such that $M(p,c)=1$. This implies that there is a complete path P through M with $\mathbb{P}(P) \subseteq U \cup W \cup S$. Together with Remark 10 this implies that there is a path $S \in \mathcal{P}$ with $\mathbb{L}(S) \subseteq \mathbb{L}(P) \subseteq U \cup W \cup S$. ■

Lemma 4.3 shows that the method of iterated consensus applied to a set $\mathcal{P}(M)$ does not change $\mathcal{P}(M)$.
Therefore we have the following

4.4 Corollary:

$\mathcal{P}(M)$ is the set of prime implicants of $F_c(M)$. ■

Since the set of prime implicants of an unsatisfiable propositional formula is empty, we have

4.5 Corollary:
Let $\mathcal{F}$ be an unsatisfiable ground formula and M the NF-matrix of $\mathcal{F}$. Then $\mathcal{P}(M)=\emptyset$. ∎

The next corollary shows that one can dispense with the deletion of absorbed columns for the price of obtaining eventually more redundant paths in the result.

4.6 Corollary:
Let A be the algorithm of chapter 4, let $\mathcal{P}(M)$ be the set of paths produced by A and let A' be the same algorithm with the exception that absorbed columns of a matrix are not deleted. Then each path generated by A' is either in $\mathcal{P}(M)$ or it is subsumed by some path in $\mathcal{P}(M)$.

Proof:
Let $\mathcal{Q}$ be the set of paths produced by A'. The construction of A and A' shows that $\mathcal{P}(M) \subseteq \mathcal{Q}$. Let $Q \in \mathcal{Q} \setminus \mathcal{P}(M)$. Hence Q cannot be a prime implicant of $F_C(M)$, but Q is implied by $F_C(M)$, hence Q is subsumed by some prime implicant $Q' \in \mathcal{P}(M)$. ∎

In general the set of prime implicants of a formula does not represent its simplest equivalent. Therefore the prime implicants can be classified into three categories [LS80]:

4.7 Definition:
(i) **Core implicants** are those implicants that cannot be obtained as a resolvent of other implicants.
(ii) **Absolutely eliminable implicants** are those that can be obtained as a resolvent of implicants, but cannot itself resolve with other implicants.
(iii) **Eliminable implicants** are those that are neither core nor absolutely eliminable implicants.
(iv) A **simplest equivalent** of a formula $\mathcal{F}$ is a minimal subset S of the set of prime implicants of $\mathcal{F}$, such that S is logically equivalent to $\mathcal{F}$.

Any simplest equivalent to a formula must contain the core implicants and a subset of the eliminable implicants. The absolutely eliminable implicants can be ignored.

4.8 Example:
The "naive" way to transform the formula $\mathcal{F} = (p \Leftrightarrow q) \wedge (p \Leftrightarrow r)$ into conjunctive normal form results in the formula $\mathcal{F}' = (-p \vee q) \wedge (p \vee -q) \wedge (-p \vee r) \wedge (p \vee -r)$. The set of prime implicants of $\mathcal{F}$ is $\mathcal{P} = \{(-p \vee q), (p \vee -q), (-p \vee r), (p \vee -r), (-q \vee r), (q \vee -r)\}$. Since each element of $\mathcal{P}$ can resolve with some other element of $\mathcal{F}$ and each element of $\mathcal{F}$ can be obtained as a resolvent of two other implicants of $\mathcal{F}$, all implicants are eliminable. The sets $\{(-p \vee q), (p \vee -r), (-q \vee r)\}$ and $\{(p \vee -q), (-p \vee r), (q \vee -r)\}$ are simplest equivalents to $\mathcal{F}$.

In the following we give a characterization of core and absolutely eliminable implicanmats in terms of paths through NF-matrices.

4.9 Lemma:
Let M be an NF matrix with $\mathbb{P}^{\pm}(M) \neq \emptyset$ and $r \in \mathbb{P}^{\pm}(M)$. Suppose that the columns of M are ordered in such

a way that

$M(r,1) = \ldots = M(r,m) = 1$

$M(-r,m+1) = \ldots = M(-r,m+n) = 1$ and

$M(r,m+n+1) = \ldots = M(r,m+n+k) = 0$.Since $r \in \mathbb{P}^{\pm}(M)$ we have $m \geq 1$ and $n \geq 1$.

Let R be any complete path of $\mathcal{P}(M)$ of the form

$R = (b_1,\ldots,b_m,c_1,\ldots,c_n,d_1,\ldots,d_k)$

and let $B=\{b_1,\ldots,b_m\}$, $C=\{c_1,\ldots,c_n\}$and $D=\{d_1,\ldots,d_k\}$.

(i) Suppose $r,-r \notin B \cup C \cup D$, $B \not\subseteq C \cup D$ and $C \not\subseteq B \cup D$.

Then there are $P,Q \in \mathcal{P}(M)$ with $R=P+_rQ$ and R does not subsume any other path of $\mathcal{P}(M)$.

(ii) If there are $P,Q \in \mathcal{P}(M)$ with $R=P+_rQ$ then $r,-r \notin B \cup C \cup D$, $B \not\subseteq C \cup D$ and $C \not\subseteq B \cup D$.

Proof:

(i) The construction of M and R shows that $P := (r,\ldots,r, c_1,\ldots,c_n,d_1,\ldots,d_k)$ and $q := (b_1,\ldots,b_m,-r,\ldots,-r,d_1,\ldots,d_k)$ are complete paths of M and $R = P+_rQ$. We only have to show that $P,Q \in \mathcal{P}(M)$. We assume the converse. Then either P or Q must be a subsumed path. W.l.o.g let P be a subsumed path.

Case I: there is a path $P' \neq R$ of $\mathcal{P}(M)$ with $P'<P$. Then $r \in \mathbb{L}(P')$, since otherwise $\mathbb{L}(P') = C \cup D \subset \mathbb{L}(R)$, i.e. $P'<R$. Let $R' = P'+_rQ$. Then $R' \leq P+_rQ = R$, which implies $R'=R$. Now we have $R=P'+_rQ$ with $P',Q \in \mathcal{P}(M)$.

Case II: $R< P$, i.e. $\mathbb{L}(R) \subseteq \mathbb{L}(P)$. Since $r \notin B$, each b_i must be in $C \cup D$, which is a contradiction to the premise.

Since $\mathcal{P}(M)$ does not contain subsumed paths the second part of (i) is shown too.

(ii) $R=P+_rQ$ is only possible for $P:=(r,\ldots,r,c_1,\ldots,c_n,d_1,\ldots,d_k)$ and $Q:=(b_1,\ldots,b_m,-r,\ldots,-r,d_1,\ldots,d_k)$. If $B \subseteq C \cup D$ or $C \subseteq B \cup D$ then $R<P$ and $R<Q$, hence $P,Q \notin \mathcal{P}(M)$. ∎

4.10 Lemma:

If $R \in \mathcal{P}(M)$ with $R = P_1+\ldots+P_n$ and $P_i \in \mathcal{P}(M)$ for each i then there are $P,Q \in \mathcal{P}(M)$ with $R=Q+P$.

Proof:

We show the lemma for n=3. The proof easily can be generalized to the case n>3.

Let $R = P_1 +_p P_2 +_q P_3$ with $P_1, P_2, P_3 \in \mathcal{P}(M)$. Then there is a $Q_1 \in \mathcal{P}(M)$ with $Q_1 \leq P_1+_pP_2$. If $Q_1 = P_1+_pP_2$ we are ready. Therefore let $Q_1 < P_1+_pP_2$. If $q \notin \mathbb{L}(Q_1)$ then $Q_1 < P_1 +_p P_2 +_q P_3 = R$, which is a contradiction. If on the other hand $w \in \mathbb{L}(Q_1)$, then $Q_1 +_q P_3 \leq P_1 +_p P_2 +_q P_3 = R$. Furthermore there must be a $Q' \in \mathcal{P}(M)$ with $Q' \leq Q_1 +_q P_3 \leq R$, from which follows $Q' = Q_1 +_q P_3 = R$. Take $Q:=Q_1$ and $P:=P_3$. ∎

4.11 Corollary:

Core implicants are those implicants C that cannot be obtained in the form C=P+Q. ∎

4.12 Theorem:

(i) Each implicant $P \in \mathcal{P}(M)$ with $\mathbb{P}^{\pm}(M) \subseteq \mathbb{P}(P)$ is a core implicant.

(ii) Each implicant $P \in \mathcal{P}(M)$ with $\mathbb{P}(P) \subseteq \mathbb{P}(M) \setminus \mathbb{P}^{\pm}(M)$ that is a resolvent of other implicants is absolutely eliminable.

Proof:

(i) Let $\mathbb{P}^{\pm}(M) \subseteq \mathbb{P}(P)$ and $P=Q+_rR$. Then $r\in \mathbb{P}^{\pm}(M)$, which implies $r\in \mathbb{P}(P)$. This is a contradiction to 4.9 (ii). Now (i) follows from 4.10.

(ii) Since $-r\notin \mathbb{L}$ for each $r\in \mathbb{L}(P)$, P cannot resolve with any other implicant. ■

4.13 Lemma:

Let $\mathcal{F}$ be a set of prime implicants. Each implicant P of $\mathcal{F}$ that is absolutely eliminable does not occur in any simplest equivalent to $\mathcal{F}$.

Proof:

Let $\mathcal{G}$ be a simplest equivalent to $\mathcal{F}$ and $p\in \mathcal{F}$ be absolutely eliminable, i.e. there are $Q,R\in \mathcal{F}$ with $P=Q+R$ and P does not resolve with any element from $\mathcal{F}$.

Now suppose $P\in \mathcal{G}$. If Q and R are in $\mathcal{G}$, then $\mathcal{G}\backslash\{P\}$ is equivalent to $\mathcal{G}$, which is a contradiction. Assume $Q\notin \mathcal{G}$. Since $Q\in \mathcal{F}$, there are $S,T\in \mathcal{G}$ with $Q=S+T$. Hence $P=S+T+R$ and again $\mathcal{G}\backslash\{P\}$ is equivalent to $\mathcal{G}$. ■

4.14 Example:

Consider the problem to transform the formula

$\mathcal{F} \equiv (p\wedge q\wedge u) \vee (-p\wedge -q\wedge r) \vee s$

into conjunctive normal form. We have the following NF matrix M:

M =

	1	2	3
p	1	0	0
-p	0	1	0
q	1	0	0
-q	0	1	0
r	0	1	0
s	0	0	1
u	1	0	0

Then $\mathcal{P}(M)$ = {(p,-q,s), (-p,q,s), (p,r,s), (-p,s,u), (q,r,s), (-q,s,u), (r,s,u)}.

We have $\mathbb{P}^{\pm}(M)$={p,q}. Lemma 4.12(i) shows that the implicants (p,-q,s) and (-p,q,s) are core implicants. All other implicants are seen to be resolvents according to 4.9(i):

$(p,r,s) = (q,r,s) +_q (p,-q,s)$

$(-p,s,u) = (-q,s,u) +_q (-p,q,s)$

$(q,r,s) = (p,r,s) +_p (-p,q,s)$

$(-q,s,u) = (-p,s,u) +_p (p,-q,s)$

$(r,s,u) = (p,r,s) +_p (-p,s,u) = (q,r,s) +_q (-q,s,u)$

Since $\mathbb{P}(M)\backslash\mathbb{P}^{\pm}(M)$ = {r,s,u}, it follows from 4.12(ii) that (r,s,u) is absolutely eliminable. Any simplest CNF equivalent to $\mathcal{F}$ consists of the core implicants (p,-q,s) and (-p,q,s), one element of the set {(p,r,s), (q,r,s)} and one element of the set {(-p,s,u), (-q,s,u)}.

Our definition of core implicants (4.7) differs from [LS80], where a core implicant is defined to be any implicant of a CNF formula that is not implied by all other implicants. We now show that the two definitions are equivalent:

4.15 Lemma:

Let $\mathcal{F}$ be a set of prime implicants of a formula $\mathcal{G}$ in conjunctive normal form. If there is a P∈ $\mathcal{F}$ such that P is implied by all other implicants of $\mathcal{F}$, then there are $Q_1,...,Q_n \in \mathcal{F}$ with $P=Q_1+...+Q_n$.

Proof:

Either P itself is a resolvent of implicants of $\mathcal{F}$ or P is subsumed by a resolvent of implicants in $\mathcal{F}$. In the first case we are ready. Now assume R < P with $R=R_1+...+R_m$ and the R_i are all in $\mathcal{F}$. This implies R'≤R for some R'∈ $\mathcal{F}$. Hence we have R'<P with R'∈ $\mathcal{F}$, which implies P∉ $\mathcal{F}$, a contradiction. ■

4.16 Theorem:

C is a core implicant of a set $\mathcal{F}$ of prime implicants of a formula $\mathcal{G}$ iff C is not implied by $\mathcal{F}\setminus\{C\}$.

Proof:

a) If C is not implied by $\mathcal{F}\setminus\{C\}$, then it cannot be obtained in the form C = P+Q. Then 4.11 implies that C is a core implicant.

b) If C is implied by $\mathcal{F}\setminus\{C\}$, then it cannot be a core implicant due to 4.15. ■

Loveland and Shostak [LS80] define absolutely eliminable implicants to be those implicants that are resolvents of core implicants. But this definition does not catch the intuitive meaning of absolutely eliminable, since it cis possible, that an implicant does not occur in any simplest equivalent to a formula but nevertheless it is not absolutely eliminable: In example 4.14 the implicant (r,s,u) does not occur in any simplest equivalent to $\mathcal{F}$ (it is absolutely eliminable according to definition 4.7) but it is not a resolvent of the core implicants (r,-q,s) and (-p,q,s).

5. Conversion of Arbitrary Formulae

In section 3 we described the basic step of the conversion algorithm: the conversion between conjunctive and disjunctive normal form. The transformation of an arbitrary formula $\mathcal{F}$ into clausal normal form (or disjunctive normal form, respectively) starts with the innermost terms of $\mathcal{F}$ and multiplies them using successively the basic algorithm until the desired normal form is achieved.

5.1 Example:

Let

$\mathcal{F} := x \vee ((y\vee(z\wedge x)) \wedge (-y\vee(z\wedge -x)).$

The transformation of $\mathcal{F}$ into CNF takes the following steps: First the innermost formulae $y\vee(z\wedge x)$ and $-y\vee(z\wedge -x)$, which are in DNF, are transformed into CNF and then concatenated. This yields:

$\mathcal{F}_1 := x \vee ((y\vee z)\wedge (y\vee x)\wedge (-y\vee z)\wedge (-y\vee -x))$

Now the innermost formula $(y\vee z)\wedge (y\vee x)\wedge (-y\vee z)\wedge (-y\vee -x)$ is transformed from CNF into DNF:

$\mathcal{F}_2 := x \vee (y\wedge z\wedge -x) \vee (-y\wedge z\wedge x)$

The final transformation is from DNF into CNF and yields

$\mathcal{F}_3 := (x \vee y) \wedge (x \vee z)$

which represents the set of prime implicants of $\mathcal{F}$.

On the other hand the outermost-innermost multiplication yields the formula

$\mathcal{G} := (x \vee y \vee z) \wedge (x \vee y \vee x) \wedge (x \vee -y \vee z) \wedge (x \vee -y \vee -x)$

which can be reduced to $(x \vee y) \wedge (x \vee -y \vee z)$ by removing subsumed clauses. The reduction to the set of prime implicants, however, can only be achieved by a resolution step.

The multiplication algorithm described in this paper has been implemented in the MKRP theorem prover [KM84]. It works especially well for examples with nested equivalences, as it is the case with Andrew's example [H80]. An equivalence of the form $f_1 \Leftrightarrow f_2 \Leftrightarrow \ldots \Leftrightarrow f_{n+1}$ will result in 2^n clauses in the best case of transformation and in 4^n clauses in the worst case. Our algorithm always produces the minimal number of 2^n clauses.

Acknowledgement

I would like to thank Jörg Siekmann and Hans-Jürgen Bürckert for their carefully reading of an earlier draft of this paper. They had some good ideas to improve the paper.

References

[EW83] Eisinger, N.; Weigele, M.: *A Technical Note on Splitting and Clausal Normal Form Algorithms.* Proc. of 7th German Workshop on Artificial Intelligence, Dassel/Solling. Springer IFB 76, (1983).

[H80] Henschen, L. et al. : *Challenge Problem 1.* SIGART Newsletter 72, 30 - 31 (1980).

[KM84] Karl Mark G. Raph: *The Markgraph Karl Refutation Procedure.* Interner Bericht, SEKI-Memo MK-84-01, Universität Kaiserslautern (1984).

[LS80] Loveland, D.W.; Shostak, R.E.: *Simplifying Interpreted Formulas.* Proc. of 5th Conference on Automated Deduction, Les Arcs, 97 - 109 (1980).

[Qu59] Quine, W.V.: *On Cores and Prime Implicants of Truth Functions.* Am. Math. Monthly, 66, 755 - 760 (1959).