

## 組合せ回路の冗長除去における含意関係の不変性について

梶原誠司, 市原英行, 樹下行三

大阪大学大学院 工学研究科 応用物理学専攻

〒565 吹田市山田丘2-1

TEL: 06-879-7835 FAX:06-879-7836

E-mail: {kajihara, icchii, kinoshita}@ap.eng.osaka-u.ac.jp

あらまし SOCRATES に基づくテスト生成アルゴリズムを用いた組合せ回路の冗長除去では、一つの冗長故障を取り除くごとに回路内の含意関係が変化することがあるため、静的学習をやり直すことが求められている。本論文では、冗長除去による回路変換における、静的学習で得られる含意関係の不変性について考察する。次に、静的学習をやり直す代わりに、回路変換前後で不変である含意関係を冗長判定に用いることで冗長除去を効率化する手法を提案する。静的学習のやり直す回数を削減することで、全体の処理時間は大幅に短縮する。ベンチマーク回路の組合せ回路的冗長を取り除く実験では、従来手法に比較して、最高60倍高速化できたことを示す。

キーワード 冗長除去, テスト生成, 論理合成, 静的学習, 組合せ回路

On Invariant Implication Relations in Redundancy Removal  
for Combinational Circuits

Seiji Kajihara, Hideyuki Ichihara, and Kozo Kinoshita

Dept. of Applied Physics, Osaka University

TEL: 06-879-7835 FAX:06-879-7836

E-mail: {kajihara, icchii, kinoshita}@ap.eng.osaka-u.ac.jp

**Abstract** In redundancy removal for combinational circuits using SOCRATES based test generation, it was required to do over static learning again whenever a redundant fault is removed because implication relations of the original circuits may change. In this paper we discuss invariance of implication relations derived by static learning in modifying circuits by redundancy removal, then propose an efficient redundancy removal method in which invariant implication relations are used for redundancy identification in spite of doing over static learning. Experimental results for benchmark circuits show that the proposed method is up to 60 times faster than the previous method.

**key words** redundancy removal, test generation, logic synthesis, static learning, combinational circuit

## 1. はじめに

組合せ回路のテスト生成の効率化に関する研究は、古くから行われているが、PODEM [1], FAN [2], SOCRATES [3] 等の優れたアルゴリズムの開発により、大幅な発展を成し遂げてきた。その成果は、テスト生成の高速化のみならず、回路の冗長判定の高速化にも貢献し、テスト生成技術を論理合成 [4] [5] [6] [7] [8] や論理検証 [9] [10] の一部として用いることも提案されてきている。

組合せ回路の論理合成に関しては、故障の検出可能性から回路の冗長性を判断でき、その冗長性を除去することで回路を単純化できる（このとき検出不可能な故障は冗長故障と呼ばれている）。テスト生成を利用した冗長除去の手法は、規模が大きな回路も扱うことができるという利点がある。一方で、一つの冗長故障を取り除くことにより、他の冗長故障が検出可能になる場合や、検出可能な故障が冗長故障になる場合があるため、一度に一つずつ冗長故障を取り除き、また、一つの冗長故障を取り除くごとにテスト生成を繰り返すことが求められていた [4]。冗長除去の処理を効率的に行う手法は、[5] および [6] で提案されている。いずれもテスト生成処理の繰り返しを少なくするため手法であるが、冗長判定に含意操作を伴うため、一つの冗長故障を取り除くごとに回路内の含意関係を調べ直す必要がある。

冗長除去にテスト生成を用いる場合、高い冗長判定能力を持ったテスト生成アルゴリズムが必要である。SOCRATES [3] は、静的学習 (static learning) と呼ばれる処理において、前もって個々のゲートの入出力関係だけからでは求まらないような信号値の含意関係を学習し、それらをテスト生成中の含意操作に用いることで、冗長判定の高速化に役立っている。静的学習の導入により、一意的に決まる信号値を多く求めることができ、冗長故障を従来とは比較にならないほど少ない回数のバックトラックで高速に指摘することができるようになった。このため、SOCRATES を基本としたテスト生成アルゴリズムがよく冗長除去に用いられている [5] [7]。静的学習が一般のテスト生成に用いられる場合には、前処理として一度だけ行えばよいから、総処理時間のうち静的学習に費やされる処理時間は非常に小さい。しかしながら、冗長除去においては、変換前の回路に対しては成立した信号値の含意関係が、回路変換後には無効となってしまう可能性がある。これら成立しなくなった含意関係が冗長判定に用いられるなら、判定結果は誤ったものになるため、変換前の回

路において得られた含意関係を一度すべて破棄し、新たに静的学習を行っていた。従って、冗長故障数が多い回路では、静的学習が何度も繰り返され、特に規模が大きな回路では、静的学習に多くの時間が費やされることになる。

本論文では、回路内の一部のゲートや信号線を除去するような回路変換において、変換前の回路に対する静的学習で得られた含意関係のうち、どのようなものが変換後の回路で無効になるかを考察する。次に、冗長除去の処理において、静的学習をやり直すことを止め、回路変換前後で不変である含意関係を冗長判定に用いることで、冗長除去を効率化する手法を提案する。一つの冗長故障から除去される信号線やゲートは、回路全体からみればごく一部であり、これらの除去により無効となる含意関係は必ずしも多くない。このため、最初に静的学習で得られた含意関係の中から無効になる可能性があるものを取り除き、不変であるものだけを用いて変換後の回路の冗長判定を行ったとしても、その冗長判定能力には大きな影響をもたらさない。処理時間の点では、静的学習を行う回数が大幅に削減できるため、全体の処理時間は大幅に短縮可能である。ベンチマーク回路の組合せ回路的冗長を取り除く実験では、従来手法に比較して、本手法は最高 60 倍の高速化を可能にしている。

以下、本論文は次のように構成される。2 章では、本論文に用いる用語の定義を行い、静的学習を冗長除去に用いるときの問題点について述べる。3 章では、ある回路変換により変換前の含意関係が成り立たなくなるのは、どのような場合に起きうるかを考察する。4 章では、静的学習の回数を少なくすることで冗長除去の処理を効率化する手法とベンチマーク回路に対する実験結果を示し、6 章で本論文のまとめを述べる。

## 2. 準備

### 2.1 諸定義

まず最初に、本論文で用いる語句を定義する。なお、本論文では、AND, NAND, OR, NOR, EXOR ゲートで構成された組合せ回路を扱うものとする。

[定義1 制御値] ゲートの一つの入力値により出力値が一意に決まる時、その入力値を制御値と呼ぶ。AND, NAND ゲートの場合には 0, OR, NOR ゲートの場合には 1 である。EXOR ゲートには存在しない。また、制御値でない信号値を非制御値と呼ぶ。

SOCRATES で行われる静的学習 (static learning) は、個々のゲートの入出力関係だけからでは求まらないよ

うな信号値の含意関係をテスト生成に前もって抽出する操作である [3]。例えば、図1に示した回路では、静的学習の結果、“ $d=1$ ならば $a=1$ ”という含意関係を学習することができる。これは、 $a=0$ からの含意操作によって得られた命題“ $a=0$ ならば $d=0$ ”の対偶を考えることによって得られる。ここでいう含意操作とは、ある信号値が決まったことから一意に決まる信号値を割り当てる操作のことである。学習した含意関係は、テスト生成の含意操作において信号線  $d$  に 1 を割り当てるときに使用することにより、一意的に決定する信号値をより多くの信号線に対して求めることが可能となる。

静的学習では、信号線  $i$  の値を  $v_i \in \{0, 1\}$  として含意操作を行い、その結果、ゲート  $G$  の出力信号線  $j$  の値が  $v_j \in \{0, 1\}$  となったとき、“ $j=v_j$ ならば $i=\bar{v}_i$ ”という信号値の含意関係を記憶するものである。記憶する含意関係は、以下に示す学習基準(1)と(2)を同時に満たすときであり、それ以外の信号線の値は、 $j=\bar{v}_j$  の含意操作から直接得られるので、記憶する必要はない。

[学習基準]

- (1) ゲート  $G$  の出力値  $v_j$  が入力の非制御値ではない。
- (2) 外部出力方向への含意操作が信号値  $v_j$  の決定に関与している。

[定義2 含意情報] 静的学習によって記憶された含意関係を含意情報と呼ぶ。

言い換えれば、含意情報とは、ある信号値から含意操作を行うことにより得られた含意関係の中で、上記の学習基準を満たしたものの対偶が示す含意関係のことである。

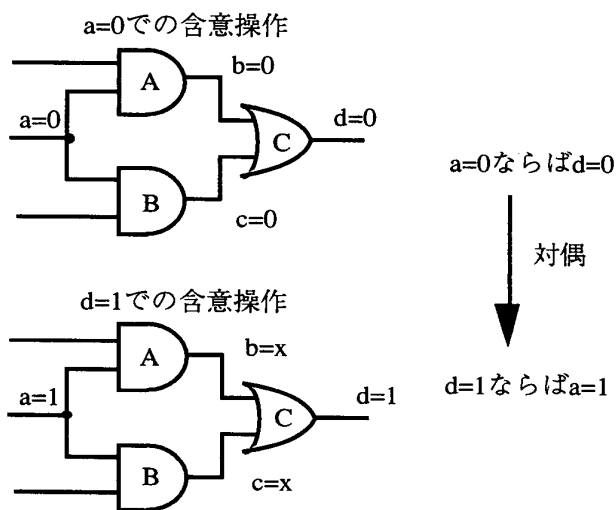


図1 静的学習による含意情報の抽出

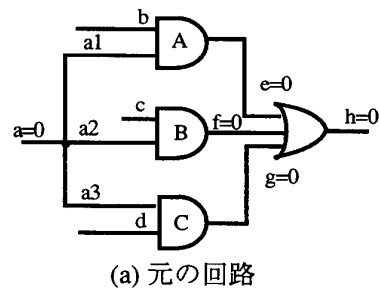
次に、含意操作を直接含意操作と間接含意操作に分類する。

[定義3 直接含意操作・間接含意操作] あるゲートまたは分岐点の入出力値の割り当てから、同じゲートまたは同じ分岐点の他の入出力値を決める含意操作を直接含意操作といい、含意情報を用いて信号値を決める含意操作を間接含意操作という。

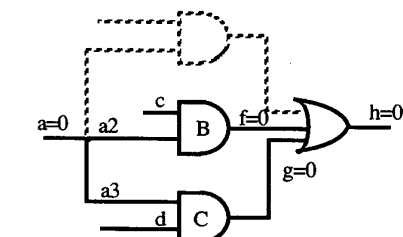
図1の例で、信号値の含意関係を学習するために行った含意操作はすべて直接含意操作である。また静的学習の結果として得られた“ $d=1$ ならば $a=1$ ”という含意情報に基づいて信号線  $a$  の値を決定する含意操作は間接含意操作である。

2. 2 冗長除去と含意情報

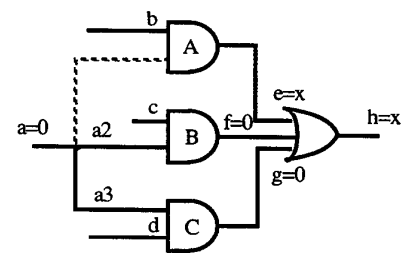
静的学習により含意情報は、ある冗長故障を除去することにより成立しなくなる場合がある。例えば与えられた回路が図2(a)のような回路を含んでいるとする。この部分回路に対して、“ $a=0$ ならば $h=0$ ”という含意関係が成立するので、“ $h=1$ ならば $a=1$ ”という含意情報が得られる。今仮に、信号線  $a1$  の縮退故障が冗長故障でありこれを除去したとすると、図2(b)



(a) 元の回路



(b) 含意情報が不変である冗長除去



(c) 含意関係が無効となる冗長除去

図2 冗長除去における含意情報の不変性

のような回路が得られる。図2(b)の回路に対しても、元の回路で得られた" $h=1$ ならば $a=1$ "という含意情報は成立する。しかしながら、信号線 $a1$ の1縮退故障が冗長故障であり、これを除去したとすると図2(c)のような回路が得られ、この回路に対しては、" $h=1$ ならば $a=1$ "という含意情報は成立しなくなる。ここでは、例を示すことは省略するが、冗長除去により、新たに含意情報が得られる場合があることを記しておく。

図3は、ある一つの冗長部分を取り除く回路変換の前後で得られる含意情報の集合の関係を示す。Aは、回路変換によって無効となる含意情報を表す部分であり、Bは回路変換後も成り立っている含意情報を示している。Cは回路変換後の回路において新たに得られた含意情報を表す。すなわち、AとBが回路変換前に有効な含意情報の集合で、BとCが回路変換後に有効な含意情報の集合である。回路変換後成り立たなくなった含意情報を用いたテスト生成は、誤った冗長判定結果をもたらすため、従来の手法において、AとBの含意情報を一度破棄し、改めて静的学習を行いBとCの含意情報を学習し直していた。

一般に、一つの冗長故障から除去される信号線やゲートの数は、数本または数個程度であり、Aの含意情報は少ないと考えられる。これは、多くの含意情報は変換後の回路においても有効であるにもかかわらず、それらを破棄して同じ含意情報を学習することに多大な時間を割いていることを意味しており、回路単純化に必要な計算時間の増加の一つの大きな原因となっている。

本論文では、Aの含意情報を完全にに取り除き、残ったBの含意情報のみを用いてテスト生成を行うことを考える。これは、Cの含意情報はあまり多くないことから、Bの含意情報だけでもテスト生成の含意操作の

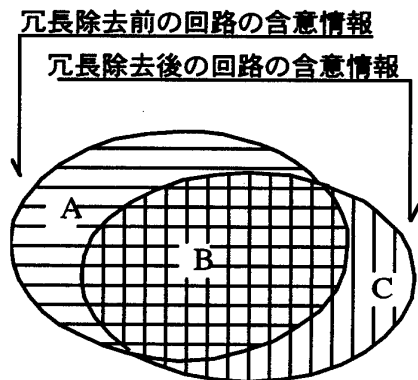


図3 冗長除去と含意情報の変化

能力向上に十分な成果があると考えられるためである。この操作により、静的学習をやり直すことは必要なくなり、計算時間の短縮が期待できる。もし、Bの含意情報だけのテスト生成において、対象故障の検出可能性の判定ができない場合には、改めて静的学習のやり直しを行うことにする。次章では、Aの部分に相当する無効となる含意情報がどのように生じるかを考察し、それらを取り除くような含意情報の部分的な変更方法について述べる。

### 3. 含意情報の不変性

#### 3.1 直接含意操作における含意関係

含意情報は、ある信号線の値に対して含意操作を行った結果得られる含意関係であるが、それが直接含意操作のみを行って得られた含意関係である場合と、直接含意操作と間接含意操作の双方を用いて得られた含意関係である場合の2通りが考えられる。直接含意操作のみにより得られた含意情報が冗長除去による回路変換により無効になるか否かは、含意情報を得るために用いた直接含意操作の含意関係が回路変換後も維持できるか否かで決まる。一方、直接含意操作と間接含意操作の双方を用いて得られた含意情報は、そのときに用いた直接含意操作と間接含意操作の双方の含意関係が維持できるか否かで決まる。静的学習の一番最初に得られる含意情報は、必ず直接含意操作のみにより得られた含意関係であることを考えれば、すべての含意情報は、直接含意操作が無効になるか否かを判断すれば、間接含意操作を伴う場合でも、回路変換後に無効になるか否かを連鎖的に判断できる。そこで、直接含意操作における含意関係が冗長除去により変わる場合を考察する。

#### 3.2 論理ゲートの入力の除去と含意関係

直接含意操作における含意は、論理ゲートの入出力値の関係から値を決める場合と、分岐点において値を決める場合があるが、まず論理ゲートの入出力値の関係から値を決める場合について考える。

ゲートの入力値が決まることにより、そのゲートの出力値が決まる場合には、入力値として制御値を持つときと、非制御値を持つときがある。図4にANDゲートの例を示す。図4(a)のように入力値として制御値を持つとき、その信号線を除去することは除去後の回路でゲートの出力値を変える可能性がある。このため、除去されたゲートの入力信号線が、含意情報を得るための含意操作で制御値を持つように値が設定されたな

ら、その含意情報は無効になる可能性を持つ。図4 (b) は、入力値として非制御値を持つ場合であるが、その信号線を除去することは除去後の回路で決してゲートの出力値を変えない。このため、含意情報を得るための含意操作で非制御値を持つように値が設定されたゲートの入力信号線を除去しても、除去後の回路におけるその含意情報には影響しない。ただし、EXORゲートは制御値を持たないものの一つの入力信号線が除去されることでその出力値が影響をうけるため、EXORゲートの入力信号線を除去した場合は含意情報は無効となる可能性を持つ。

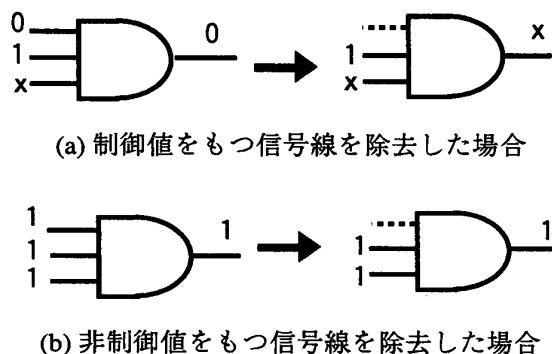


図4 冗長除去とゲートの出力の変化

ゲートの出力信号線が除去される場合、そのゲート及びゲートのすべての入力信号線が除去されるので、ゲートの出力値が決まることによりそのゲートの入力値が決まる含意操作はここで考慮する必要はない。

### 3. 3 分岐の枝の除去と含意関係

分岐点における含意操作には、ある分岐の枝の値が決まった後、分岐の幹及び同じ分岐点の他の分岐の枝の値が決まる場合と、分岐の幹の値が決まった後、その分岐の枝の値が決まる場合がある。

図5 (a) はある分岐の枝の値が最初に決まった場合の例を示すが、その分岐の枝が除去されたとしても、除去後の回路において、残った分岐の枝や分岐の幹の値は決定しない。つまり、ある含意情報を得るときにこのような含意操作が行われている場合には、その含意情報は無効になる可能性がある。図5 (b) は分岐の幹の値が最初に決まる例を示すが、この場合、ある分岐の枝が除去されたとしても、除去後の回路において、残った分岐の枝や分岐の幹の値は変わらない。また、分岐の幹が除去される場合は、その分岐のすべての枝の信号線が除去されるのでここで考慮する必要はない。つまり、含意情報を得るための含意操作で最初に

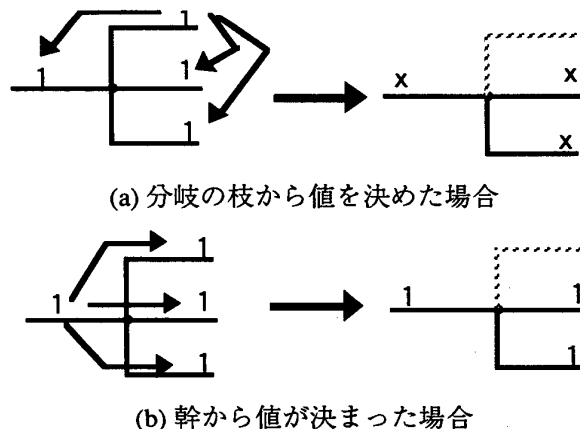


図5 冗長な分岐の枝の除去と信号値の変化

分岐の幹に値が設定された場合には、除去後の回路におけるその含意情報には影響しない。

### 3. 4 無効となる可能性をもつ含意情報

3.1節から3.3節までの含意情報の不変性の考察に基づいて、無効となる可能性をもつ含意情報について述べる。これは、除去した信号線に対し、除去前の回路のある含意情報を得るときに行った含意操作でどのような信号値を与えたかを調べることで判断できる。このとき除去したすべての信号線について調べる必要はない。

ゲートの入力信号線を調べる場合、同じゲートの出力信号線が除去された場合は考える必要がないので、結局冗長部分の先頭の信号線だけを考えればよいことになる。なぜなら、先頭の信号線から外部入力方向に向かって外部入力および分岐点までの信号線は除去されているので、他に入力信号線だけが除去されているようなゲートは存在しないからである。分岐点についても同様な考察から、枝のみが除去され幹が残っている分岐点で調べればよいことになる。

以上の考察から、ある含意情報が無効となる可能性をもつことになるのは、その含意情報を得るための含意操作において、

- ・ 除去部分の先頭の信号線に制御値を割当てた場合、
  - ・ 除去部分の先頭の信号線が EXOR ゲートの入力信号線の場合、
  - ・ 除去した分岐の枝から分岐の他の枝や幹の信号値が決まった場合、
  - ・ 間接含意操作を介しているならば、その間接含意操作に用いた含意情報そのものが無効である場合、
- のいずれかに該当するときである。

## 4. 実験結果

### 4.1 アルゴリズム

図6に本手法を用いた回路簡単化のアルゴリズムを示す。これは、検出不能故障のクラス化による組合せ回路の冗長除去の方法 [5] を基本としたアルゴリズムであり、太い線で囲った部分が新しく導入した部分である。"付加情報の記憶"は、静的学習時に含意情報を学習するたびに、冗長除去後の含意情報の有効性を判断するときに必要な付加情報を記憶する処理である。また、"含意情報の部分的な変更"では、記憶した付加情報を用いて、無効となる可能性のある含意情報の除去操作を行う。テスト生成の後の分岐は、テスト生成に失敗したときに、再学習を行うためのものである。テスト生成の失敗とは、3回以下のバックトラックでテストパターンを生成できなかったことを意味する。

本手法では、テスト生成に失敗したときには改めてすべての含意情報を除去し学習をやりなおす再学習をアルゴリズムに加えている。これは無効となる可能性

のある含意情報を除去しただけでは、テストパターン生成に失敗する恐れがあるためである。この理由は、静的学習をやり直すことで発見が可能であった含意情報 (図3のCにあたる部分) が得られないこと、あるいは、実際は有効なままである含意情報 (図3のBにあたる部分) が除去されることが挙げられる。

### 4.2 実験結果

上記のアルゴリズムを富士通 S-4/LC ワークステーション上に C 言語を用いてプログラム化し、ISCAS'85 のベンチマーク回路 [11] と ISCAS'89 のベンチマーク回路 [12] の組合せ部分を用いて実験した。本手法の有効性を示すため、静的学習を回路変換の度にやり直す冗長除去の手法 [5] と比較したが、その結果を表1に示す。まず、表中の "A: 従来法" の "時間" 欄には、[5] の手法による実行時間を示している。"静的学習" の "実行回数" と "時間" の欄は、[5] の手法において静的学習が行われた回数と一回目の静的学習に費やした時間を示している。静的学習の "実行回数" と "時間" に示す値の積が、"A: 従来法" の処理時間のうち静的学習に費やした時間とほぼ等しく、この時間を短くすることが本手法の目的である。"B: 提案手法" の "時間" の欄は、本手法の実行時間を示しており、"加速率" の欄には、"A: 従来法" と比較して何倍高速になったかを示している。規模が小さい回路では、1回の静的学習に必要な時間が短く、また、静的学習の繰り返し回数が少ないため、本手法の有効性は余り認められない。逆に、1回の静的学習に必要な時間が長く、また、静的学習の繰り返し回数が多い回路では、本手法の有効性が顕著に現れている。特に、s35932 では約60倍も高速に処理されている。唯一、c6288 に対しては、本手法が従来手法より多くの時間を要しているが、これは含意情報の有効性を判断するときに必要な付加情報の計算と記憶のため、より多くの時間を費やしたことが理由として考えられる。

最後に、結果として示していないが、提案手法と従来手法では、除去される故障の順序が異なる場合があるため、結果として得られる回路は違うものになる可能性があるが、除去された信号線数やゲート数の差は、どの回路においても非常に小さく、この観点から手法の優劣を論ずることはできなかった。また、提案手法において、新たに生じる含意情報をつけ加えていないためテスト生成に失敗して、再学習を必要とした回路は一つもなかった。

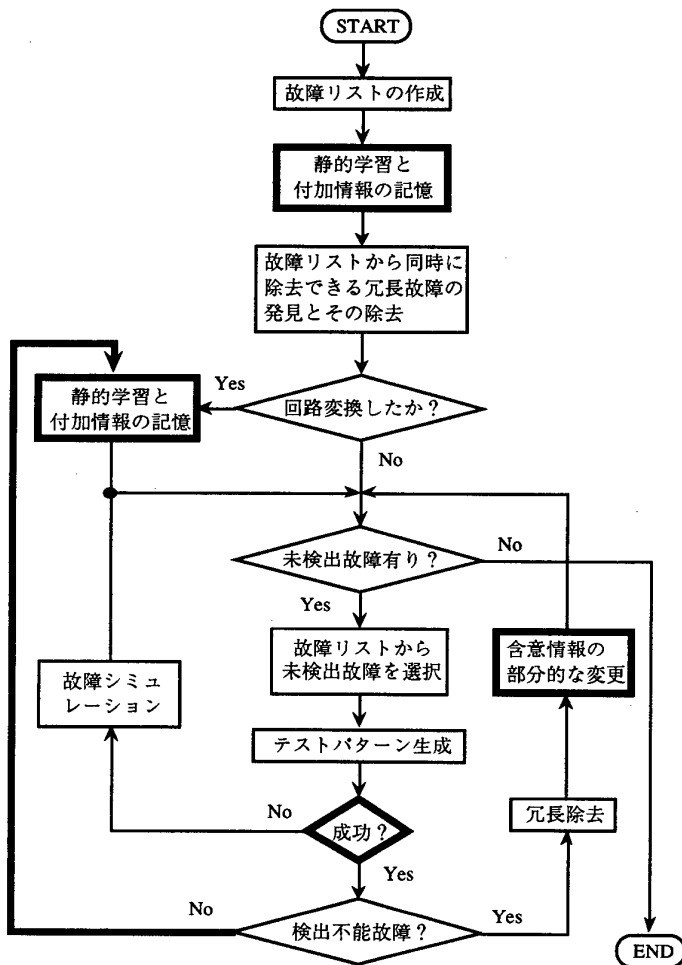


図6 回路簡単化のアルゴリズム

表1 ベンチマーク回路に対する実験結果

回路	A: 従来法	静的学習		B: 提案手法	加速率 (A/B)
	時間(sec)	実行回数	時間(sec)	時間(sec)	
c432	3	5	0.1	3	1.0
c499	4	9	0.1	3	1.3
c880	3	1	0.1	3	1.0
c1355	20	9	0.5	17	1.2
c1908	17	9	0.5	16	1.1
c2670	31	41	0.6	25	1.2
c3540	95	37	1.9	51	1.9
c5315	124	58	1.1	81	1.5
c6288	68	3	0.9	91	0.7
c7552	786	139	2.8	437	1.8
s9234	3445	169	4.6	2887	1.2
s13207	4956	383	13.2	917	5.4
s15850	2806	294	8.3	725	3.9
s35932	73635	657	77.3	1221	60.3
s38417	2995	70	9.1	2898	1.0
s38584	27875	254	110.6	4670	6.1

## 5. まとめ

本論文では、冗長除去に対する含意情報の不変性について考察を行い、含意情報の不変性を利用した静的学習のやり直しを行わない冗長除去手法を提案した。ベンチマーク回路に対しての実験では、従来法と比べ最高60倍もの高速化できたことを示した。今回は回路内の冗長部分を見つけて除去する回路変換しか扱わなかったが、今後の課題として、信号線を付加することにより新たに冗長部分を作り出して除去する論理合成手法にも対応させることが考えられる。

## 参考文献

- [1] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. Comput.*, C-30, 3, pp. 215-222 (Mar. 1981)
- [2] H. Fujihara and T. Shimono, "On the Acceleration of Test Generation Algorithms," *IEEE Trans. Comput.*, C-32, pp. 1137-1144 (Dec. 1983)
- [3] M. H. Schultz, E. Trischler and T. Sarfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," *IEEE Trans. CAD.*, pp.126-137 (Jan. 1988)
- [4] D. Bryan, F. Brglez and R. Lisanke, "Redundancy Identification and Removal," *MCNC Workshop on Logic Synthesis* (May 1989)
- [5] Kajihara, S. Shiba, H. and Kinoshita, K., "Removal of Redundancy in Logic Circuits under Classification of Undetectable Faults," *22th Int'l Sympo. on Fault-Tolerant Comput.*, pp. 263-270 (July 1992)
- [6] M. Abramovici, M. A. Iyer, "One-Pass Redundancy Identification and Removal," *Proc. Int'l Test Conf.*, pp. 807-815 (Sep. 1992)
- [7] R. Jacoby, P. Moceyunas, H. Cho and G. Hachtel, "New ATPG Techniques for Logic Optimization," *Int'l Conf. on CAD*, pp. 548-551 (Nov. 1989)
- [8] W. Kunz, P. R. Menon, "Multi-Level Logic Optimization by Implication Analysis," *Int'l Conf. on CAD* (Nov. 1994)
- [9] D. Brand, "Verification of Large Synthesized Designs," *Int'l Conf. on CAD*, pp. 534-537 (Nov. 1993)
- [10] W. Kunz, "HANNIBAL: An Efficient Tool for Logic Verification Based on Recursive Learning," *Int'l Conf. on CAD*, pp. 538-543 (Nov. 1993)
- [11] F. Breglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," In *IEEE Int'l Sympo. on Circuits and Systems; Special Session on ATPG and Fault Simulation*, pp. 663-698 (June 1985)
- [12] F. Breglez, D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," In *Proc. IEEE Int'l Sympo. on Circuits and Systems*, pp. 1929-1934 (May 1989)