論　文

探索状態被覆性に基づく探索空間削減の一手法

正　員　藤野　貴之†　　正　員　藤原　秀雄†

A Method of Search Space Pruning Based on Search State Dominance

Takayuki FUJINO † *and* Hideo FUJIWARA †, *Members*

あらまし　Giraldi と Bushnell によって，従来の手法とは異なった観点より探索空間を削減する，EST (Equivalent STate hashing) アルゴリズムが提案された．これは，過去に行われたテストパターン探索の経過および結果を探索履歴として記憶し，その履歴を用いることで無駄な探索を避け，探索空間を減らすものである．本論文では，EST で新しく導入された探索状態等価性の概念を拡張した探索状態被覆性を新たに定義する．更に，探索状態被覆性に基づく DST (Dominant STate hashing) アルゴリズムを提案する．探索状態被覆性は，探索状態等価性に比べ更に探索空間を削減することが可能である．

キーワード　組合せ論理回路，探索空間削減，探索状態被覆性，探索履歴，テスト生成

1. まえがき

高度に発達を続ける回路集積化技術により，ますます大規模化する VLSI 回路の自動テスト生成において要求される課題は，高い故障検出率を有するテスト系列を高速に求めるテスト生成アルゴリズムの研究開発である[1]．テスト生成アルゴリズムとして最初に考案された D アルゴリズム[3] 以来，PODEM[4]，FAN[5] とアルゴリズムは高速化かつ効率化されてきたが，ISCAS '85 のベンチマーク回路[2] において，バックトラックの多発により所定の計算時間内でテストパターン生成をすることができない打切故障が存在していた．

その後 ISCAS '85 のすべてのベンチマーク回路で打切故障をなくし，所定の計算時間以内ですべての故障についてテストパターン生成および冗長故障の判定が可能なアルゴリズムが発表されている[6]~[8]．その一つ SOCRATES[6] は，FAN の含意操作，一意活性化操作等を拡張し，新たに静的学習やテスト生成中の動的学習の操作を導入することにより，バックトラックの発生回数削減すなわち探索空間の削減に成功している．

一方，全く別の観点から探索空間を減らす方法が Giraldi と Bushnell により発表されている．EST (Equivalent STate hashing) アルゴリズム[8],[9] と名づけられたこの方法は，テストパターン探索の経過および結果を探索履歴として記憶しておき，以後の探索においてその履歴を利用することで同じ探索を極力避け，探索空間を減らすものである．EST アルゴリズムでは評価フロンティア，探索状態等価性等の概念が導入されている．

本論文では，はじめに EST アルゴリズムで導入された評価フロンティアおよび探索状態等価性について解説する．続いて，探索状態等価性を拡張した探索状態被覆性を新たに定義し，その探索状態被覆性に基づく DST (Dominant STate hashing) アルゴリズムを提案する．実験結果では，探索状態被覆性が探索状態等価性よりも更に探索空間を削減していることを示す．

2. EST アルゴリズム

2.1 探索状態と評価フロンティア

テストパターン生成問題は，2 分決定木で表現される探索空間を探索する問題として定式化される．図 1 において，各枝は回路の信号線への値の割当てを示しており，実線の枝は既に値が割り当てられ探索済みであることを，点線の枝は未探索であることを示している．各節点は，割り当てられた値をもとに含意操作が

† 明治大学理工学部情報工学科，川崎市
School of Science and Technology, Meiji University, Kawasaki-shi, 214 Japan

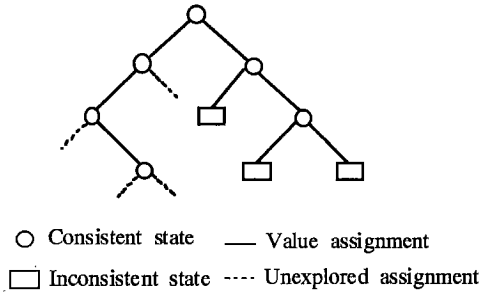


図1　テストパターン探索を表す2分決定木
Fig. 1　Decision tree.

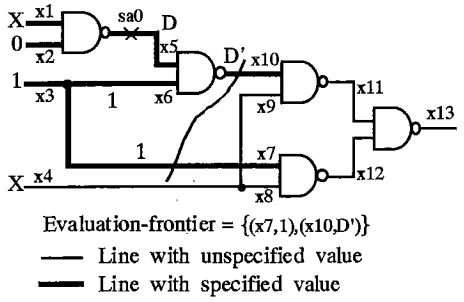


図2　評価フロンティアを用いた探索状態の表現
Fig. 2　Search state representation using evaluation-frontier.

実行された結果生じた回路の状態を表現している．これを探索状態と呼ぶ．丸い節点は無矛盾な状態であることを，四角い節点は含意操作の結果矛盾が生じたか，あるいはDフロンティアが消失したことによって，バックトラックが発生したことを示している．

信号線への値の割当ては，テスト生成に用いられるアルゴリズムに依存するが，例えばPODEMを考えると，回路の状態は外部入力からの含意操作により一意的に決定されるので探索状態は外部入力の状態によって表現することができる．一方，FAN，SOCRATES等のアルゴリズムは内部信号線にも値を割り当てるので，より一般的な探索状態の表現法が必要となる．

GiraldiとBushnellは，探索状態を表現する手段として評価フロンティアと呼ばれるものを導入している．これは図1のような2分決定木の各節点において，含意操作の結果回路が値の定められた既定部と，値が特定されない未定部に分割可能であることを利用したもので，既定部と未定部の境界となる信号線およびその信号値の集合によって探索状態を表現するものである．GiraldiとBushnellは，評価フロンティアを以下のように定義している．

（1）　回路を分割するカットセットとなる信号線
（2）　値が割り当てられている（すなわち，値はXでない）
（3）　外部出力へのXパスが存在する．

以上の三つの条件を満たす，信号線およびその信号値の対の集合．但し，内部信号線に値を割り当てるアルゴリズムの場合，更に(1)，(2)の条件を満たす未正当化信号線が付加されたものになる．記述形式は$\{(s,v) \mid s$は信号線名，vは信号値$\}$ないしは$\{s=v \mid s$は信号線名，vは信号値$\}$となる．評価フロンティアは，回路分割を一意に表現することが可能である．

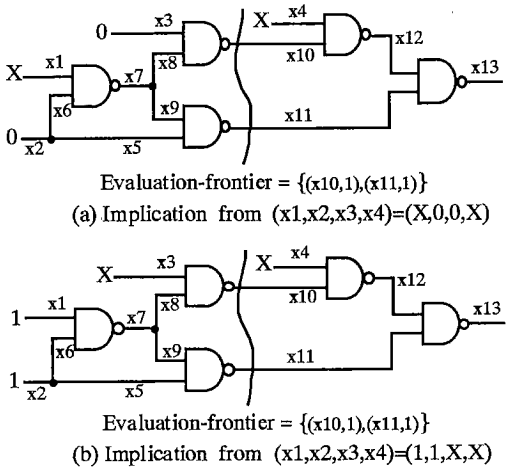


図3　等価な探索状態
Fig. 3　Equivalent search state.

図2において，外部入力$(x1,x2,x3,x4)=(X,0,1,X)$による含意操作の結果生じた探索状態は，評価フロンティア$\{(x7,1),(x10,D')\}$によって表現される．

2.2　探索状態等価性を用いた探索空間の削減

評価フロンティアは回路分割を一意的に表現できるが，ある割当てからの含意操作の結果得られた探索状態を表現する評価フロンティアと，異なった割当てからの含意操作の結果得られた評価フロンティアが等しくなることがある．例えば，図3において，(a)，(b)はそれぞれ異なった外部入力割当てから含意操作を行っているが，評価フロンティアはどちらも$\{(x10,1),(x11,1)\}$となる．このように評価フロンティアが同じになる二つの探索状態を等価な探索状態と言う．テストパターン探索の過程において得られた探索状態Aが，過去に行われた探索の過程で得られた探索

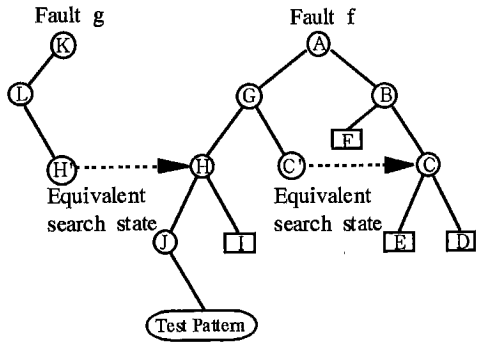


図4 探索状態等価性を用いたテスト生成
Fig. 4 Test pattern generation using search state equivalence.

状態 A' と等価である場合，探索状態 A 以降の探索結果は探索状態 A' 以降の探索結果と等しくなる．これを探索状態等価性と呼ぶ．探索状態等価性を用いることにより，等価な探索状態が現れたとき，過去の探索結果をそのまま利用することができる．結果として同じことを繰り返す無駄な探索を省くことができるので，探索空間を縮小することができる．

[例1] 図4は，ある回路における PODEM を用いたテストパターン探索の一部を2分決定木で表現したものである．はじめに，故障 f に対するテストパターン探索を考える．探索状態 A から始まり，B，C を経て，D，E，F で矛盾が発生して，各々バックトラックする．その後 G を通り，現在の探索状態は C' であるとする．ここで，探索状態 C' が既に探索した状態 C と等価である場合，すなわち C' と C の評価フロンティアが等しい場合，C' より下方の探索は行ってもむだであることがわかる．従って，これを省略し，H へと進む．これにより，無駄な探索を回避できる．図では，更に探索状態 I で矛盾が発生しバックトラックの後に，J を経てテスト生成に成功している．生成されたテストパターンは $(x1,x2,x3,x4)=(0,0,0,1)$ である．

次に，故障 g に対するテストパターン探索を考える．探索状態 H' において，先の故障 f の探索状態 H と H' が等価である場合，ここで故障 f に対するテストパターンを用いて故障 g に対するテストパターンを生成し，テスト生成を終了することができる．故障 g のテストパターンは，探索状態 H' において値の割り当てられていない外部入力に，故障 f に対するテストパターンの対応する値を割り当てたものである．従って，$(x1,x2,x3,x4)=(1,0,0,1)$ となる． □

3. 探索状態被覆性

ここで，Giraldi と Bushnell の提案した探索状態等価性を拡張した探索状態被覆性を定義する．探索状態等価性は，等価な関係にある評価フロンティアによってそれぞれ表現される回路の状態，すなわち等価な探索状態において，一方が他方の探索履歴を利用するものである．同様に，探索状態被覆性は，被覆関係にある評価フロンティアによってそれぞれ表現される二つの探索状態において，一方が他方の探索履歴を利用するものである．

3.1 評価フロンティアの被覆

先に述べたように，評価フロンティアは信号線とその値の対の集合で表現される．信号値は，正常信号値 $(0,1)$ もしくは故障信号値 (D,D') をとるので，評価フロンティア E_i は，正常信号値をもつ部分集合 E_{in} と故障信号値を持つ E_{if} に分けることができる．

ここで，次のような部分集合を考える．

$E_{if}'=\{(s,v)\mid s$ は信号線名，v は E_{if} に含まれる s に対応した信号線の故障信号値を反転したもの$\}$

例えば，評価フロンティア $E_i=\{(x1,1),(x3,1),(x9,D),(x11,D')\}$ について，$E_{if}=\{(x9,D),(x11,D')\}$ となり $E_{if}'=\{(x9,D'),(x11,D)\}$ となる．

評価フロンティアに D，D' の故障信号値が含まれるとき，すなわち $E_{if}\neq\{\phi\}$ であるとき，これを活性化された評価フロンティアと呼ぶ．

[定義1] 評価フロンティア E_i と E_j について，

(1) $(s,v)\in E_{in}\Rightarrow(s,v)\in E_{jn}$

(2) $(s,v)\in E_{if}\Leftrightarrow(s,v)\in E_{jf}$ または
$(s,v)\in E_{if}\Leftrightarrow(s,v)\in E_{jf}'$

であるとき，E_i は E_j を被覆すると言う． □

[例2] 四つの評価フロンティア $E_1=\{(x1,0),(x2,1)\}$，$E_2=\{(x1,0),(x2,1),(x3,1),(x4,0)\}$，$E_3=\{(x1,0),(x2,1),(x3,1),(x4,D')\}$，$E_4=\{(x1,0),(x3,1),(x4,D)\}$ について考える．E_1 と E_2 は(1)，(2)の条件を満たしているので，E_1 は E_2 を被覆している．しかし，E_1 と E_3 については(1)の条件は満たしているが(2)の条件を満たさないので，被覆関係は成立しない．E_3 と E_4 については被覆関係が成立する． □

評価フロンティア E_A，E_B によってそれぞれ表現される二つの探索状態 A，B について，E_A と E_B が被覆の関係 (E_A が E_B を被覆か，または E_B が E_A を被覆) にあるとき，探索状態 A，B は被覆の関係にあると言う．

また，判定木において，ある評価フロンティアによって表現される節点（探索状態）の下位に求めるテストパターンが存在するとき，この評価フロンティアは解をもつと言う．

3.2 探索状態被覆性を用いた探索空間の削減

テストパターン探索の過程において，ある探索状態 A が，過去においてなされた探索で得られた探索状態 A' と被覆関係にある場合，過去の探索履歴を用いて現在探索中のテストパターンに関する探索空間を削減することができる．これを探索状態被覆性と呼ぶ．

以下に探索状態被覆性に関する二つの定理を示す．

[定理1] 目標故障が同じ評価フロンティア E_i と E_j において，E_i が E_j を被覆し，E_i が解をもたないならば，E_j も解をもたない． □

(証明) 判定木において E_i，E_j に対応する節点（探索状態）をそれぞれ N_i，N_j とする．また，N_i，N_j を導き出す信号線割当てをそれぞれ A_i，A_j とする．

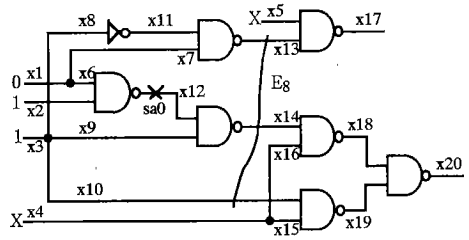
ここで，次のような集合を考える．

X_i'＝{探索状態 E_i において，X パスをもたないすべての信号線の集合}

E_i が E_j を被覆しているのであるから，E_{if} に含まれる信号線と E_{jf} (E_{jf}') に含まれる信号線は等しく，また $X_i' \subseteqq X_j'$ である．割当て A_j のうち X_i' に含まれるものは，X_i' に含まれない信号線には影響を及ぼさない．従って，A_j のうち ($X_j'-X_i'$) に含まれる割当てを N_i に付加することによって，その探索状態を $E_{jn} \cup E_{jf}$ ($E_{jn} \cup E_{jf}'$) で表現されるような節点 N_j' に到達することができる．この割当てを A_{ij} とする．

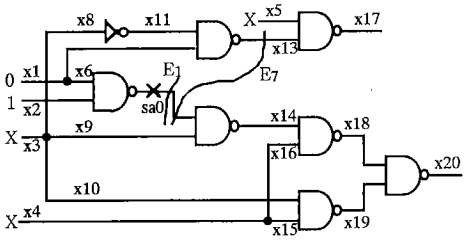
ここで，E_i は故障 f に対する解がなく，E_j は故障 f に対する解があると仮定する．E_j は故障 f に対する解があるので，判定木において節点 N_j から解の節点までの経路が存在する．節点 N_j から解（テストパターン）を導き出す割当てを A_{jt} とすると，結果として得られるテストパターンは割当て $A_j \cup A_{jt}$ より含意操作を行った後の外部入力線の信号値となる．(PODEM のように直接外部入力に値を割り当てるアルゴリズムの場合 $A_j \cup A_{jt}$ がそのままテストパターンとなる)．

節点 N_j' において得られた評価フロンティアが $E_{jn} \cup E_{jf}$ である場合，それは E_j と等価な評価フロンティアであるので，探索状態等価性を用いることにより，節点 N_j' に割当て A_{jt} を付加することでテストパターンが得られる．テストパターンが得られるということはすなわち，E_{jf} に含まれる故障信号の少なくとも一つが割当て A_{jt} によって外部出力に伝搬されるということである．



E_8＝{(x10,1),(x13,1),(x14,D')}

(a) Search state of node 8 in Figure 6(b)



E_1＝{(x12,D)}
E_7＝{(x12,D),(x13,1)}

(b) Search state of node 1 and node 7 in Figure 6(c)

図5 $x12$ 上の0縮退故障に対するテスト生成
Fig. 5 Test generation for $x12$ sa0 fault.

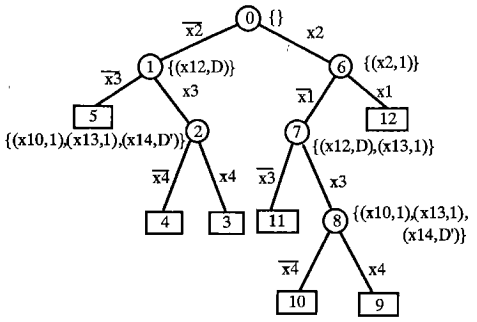
ここで，節点 N_j' において得られた評価フロンティアが $E_{jn} \cup E_{jf}'$ である場合を考える．E_{jf}' は定義より E_{jf} に含まれる故障信号値がそれぞれ反転したものであり，含まれる信号線は同じなので，E_j に含まれる故障信号の少なくとも一つが割当て A_{jt} によって外部出力に伝搬されたのであるなら，E_{jf}' に含まれる対応した信号線の故障信号は，同じ割当て A_{jt} によって外部出力まで伝搬が可能である．

従って，節点 N_j' からテストパターンまでの経路が存在する．N_j' は N_i に A_{ij} を付加することで導き出されるので，N_i もまたテストパターンまでの経路をもつ．これは，E_i が故障 f に対する解がないとした仮定に矛盾する．

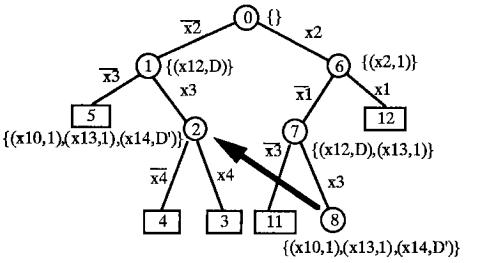
従って，E_i が E_j を被覆し E_i が解をもたないならば，E_j も解をもたない．

[例3] 図5に示す回路について，信号線 $x12$ 上の0縮退故障に対するテスト生成について考える．ここで，もとになるテスト生成アルゴリズムとして，PODEM を用いることにする．

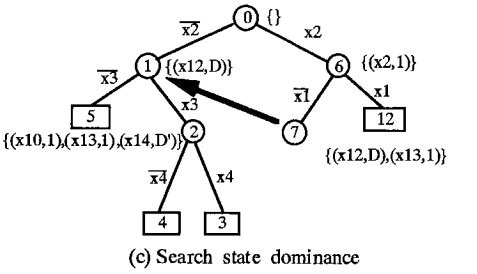
図6(a)，(b)，(c)は，テストパターン探索の過程を2分決定木で表現したものである．図6(a)は PODEM のみを用いてテストパターン探索を行った場



(a) Conventional search (using PODEM)



(b) Search state equivalence



(c) Search state dominance

図6 探索状態等価性と探索状態被覆性の比較
Fig. 6 Comparison of search state equivalence and dominance.

合である．12の節点を通過した後，目標故障が冗長故障であると判定している．図6(b)はPODEMに加えてGiraldiとBushnellの提案した探索状態等価性を適用した場合である．節点8で得られた評価フロンティア $\{(x10,1),(x13,1),(x14,D')\}$ と節点2で得られた評価フロンティアが等価であるので，探索状態等価性を用いることによって，節点8より下方の無駄な探索を避けることができる．探索状態等価性を用いることによって，PODEMと比較して2ステップ分の探索空間を削減している．図6(c)は，PODEMに加えて本論文で提案する探索状態被覆性を適用したものである．節点7において得られた評価フロンティア $\{(x12,D),(x13,1)\}$ は節点1の評価フロンティア $\{(x12,D)\}$ に被覆される．また，節点1より下方に解は存在しない．従って定理1を適用し，節点7より下方の無駄な探索を回避することができる．探索状態被覆性を適用した場合，探索状態等価性を用いた場合と比較して2ステップ，PODEMと比較して4ステップ分の探索空間の削減に成功している． □

[定理2] 目標故障が異なる活性化された評価フロンティア E_i と E_j において，E_i が E_j を被覆し，

(a) E_i が解をもたないならば，E_j も解をもたない．

(b) E_j が解をもつならば，E_i も解をもつ． □

(証明) (a)と(b)は対偶の関係にあるので，(a)が正しければ，(b)も正しい．ここでは，(b)について証明する．

判定木において E_i，E_j に対応する節点(探索状態)をそれぞれ N_i，N_j とする．また，N_i，N_j を導き出す信号線割当てをそれぞれ A_i，A_j とする．

ここで，次のような集合を考える．

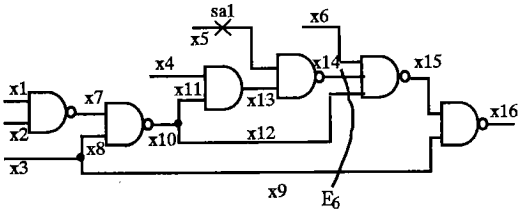
X_i' = {探索状態 E_i において，X パスをもたないすべての信号線の集合}

E_i が E_j を被覆しているのであるから，E_{if} に含まれる信号線と E_{jf} (E_{jf}') に含まれる信号線は等しく，また $X_i' \subseteq X_j'$ である．割当て A_j のうち X_i' に含まれるものは，X_i' に含まれない信号線には影響を及ぼさない．従って，A_j のうち $(X_j'-X_i')$ に含まれる割当てを N_i に付加することによって，その探索状態を $E_{in} \cup E_{if}$ ($E_{in} \cup E_{if}'$) で表現されるような節点 N_j' に到達することができる．この割当てを A_{ij} とする．

E_j は故障 f_j に対する解があるので，判定木において節点 N_j から解(テストパターン)までの経路が存在する．節点 N_j からテストパターンを導き出す割当てを A_{jt} とすると，結果として得られるテストパターンは割当て $A_j \cup A_{jt}$ より含意操作を行った後の外部入力線の信号値となる．

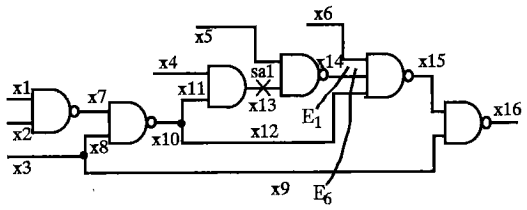
節点 N_j' において得られた評価フロンティアが $E_{jn} \cup E_{jf}$ である場合，それは E_j と等価な評価フロンティアであるので，探索状態等価性を用いることにより，節点 N_j' に割当て A_{jt} を付加することでテストパターンが得られる．テストパターンが得られるということはすなわち，E_{jf} に含まれる故障信号の少なくとも一つが割当て A_{jt} によって外部出力に伝搬されるということである．

ここで，節点 N_j' において，得られた評価フロンティアが $E_{jn} \cup E_{jf}'$ である場合を考える．E_{jf}' は定義より E_{jf} に含まれる故障信号値がそれぞれ反転したものであ



Step	Assignment	Evaluation-frontier	
1	x5=0	{(x5,D')}	
2	x3=0	{(x16,1)}	Backtrack
3	x3=1	{(x5,D'),(x8,1),(x9,1)}	
4	x1=1	{(x1,1),(x5,D'),(x8,1),(x9,1)}	
5	x2=1	{(x5,D'),(x9,1),(x11,1),(x12,1)}	
6	x4=1	{(x9,1),(x12,1),(x14,D))}	
7	x6=1	{(x16,D)}	Test generated

(a) Test generation for x5 sa1 fault



Step	Assignment	Evaluation-frontier	
1	x4=0	{(x13,D')}	
2	x5=1	{(x14,D)}	Dominates to step6 in (a)
3	x3=0	{(x16,1)}	Backtrack
4	x3=1	{(x8,1),(x9,1),(x14,D)}	
5	x1=1	{(x1,1),(x8,1),(x9,1),(x14,D)}	
6	x2=1	{(x9,1),(x12,1),(x14,D)}	Equivalent to step6 in (a)

(b) Test generation for x13 sa1 fault

図7　$x5$ 上の1縮退故障および $x13$ 上の1縮退故障に対するテスト生成

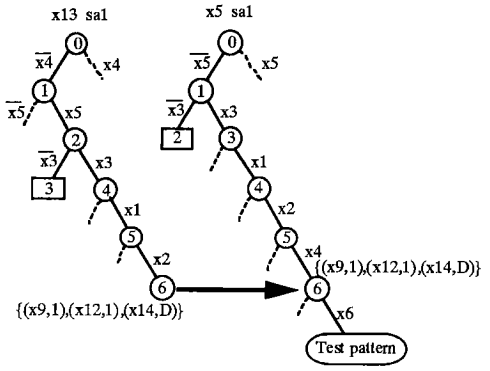
Fig. 7　Test generation for (a) $x5$ sa1 and (b) $x13$ sa1 faults.

り，含まれる信号線は同じなので，E_j に含まれる故障信号の少なくとも一つが割当て A_{jt} によって外部出力に伝搬されたのであるなら，$E_{jf'}$ に含まれる対応した信号線の故障信号は，同じ割当て A_{jt} によって外部出力まで伝搬が可能である．

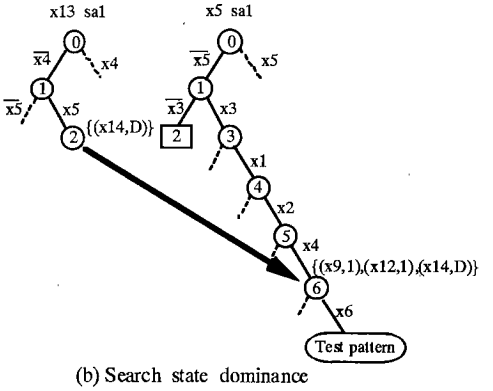
従って，節点 N_j' からテストパターンまでの経路が存在する．N_j' は N_i に A_{ij} に付加することで導き出されるので，N_i もまたテストパターンまでの経路をもつ．

従って，目標故障の異なる活性化された評価フロンティア E_i と E_j において，E_i が E_j を被覆し，E_j が解をもつならば，E_i も解をもつ．

ここで，E_i の得る解すなわち故障 f_i に対するテストパターンについて考える．



(a) Search state equivalence



(b) Search state dominance

図8　探索状態等価性と探索状態被覆性の比較

Fig. 8　Comparison of search state equivalence and dominance.

次のような集合を考える．

$I_i=\{(s,v)\mid(s,v)\in A_i$ であり，s は外部入力線，v はその値$\}$

探索状態 E_j は A_j によって得られ，更に A_{jt} を付加することで解が得られることから，テストパターンは $I_j\cup I_{jt}$ で表現することができる．探索状態 E_i を経て解を得るための割当ては $A_j\cup A_{ij}\cup A_{jt}$ であるから，テストパターンは $I_i\cup I_{ij}\cup I_{jt}$ で表現される．ここで，$A_{ij}\subseteq A_j$ であり，A_{ij} は $(X_j'-X_i')$ に含まれる割当てである．従って，$I_i\cup I_{ij}\cup I_{jt}$ は探索状態 E_i が得られた時点で値の割り当てられていない外部入力に，E_j を経て得られたテストパターンにおいて対応する外部入力線のもつ値をそれぞれ割り当てたものである．　□

[例4]　図7に示す回路について，信号線 $x5$ 上の1縮退故障（以下 $x5$ sa1 と記す）および信号線 $x13$ 上の1縮退故障（以下 $x13$ sa1）に対するテスト生成について考える．ここで，もとになるテスト生成アルゴリズムとして PODEM を用いることにする．

図8(a), (b)はテストパターン探索の過程を2分決定木で表現したものである．はじめに，$x5$ sa1 に対するテスト生成について考える．図8(a), (b)のそれぞれ右側に示されるように，$x5$ sa1 に対するテストパターンは $(x1,x2,x3,x4,x5,x6)=(1,1,1,1,0,1)$ である．続いて，$x13$ sa1 に対するテスト生成について考える．図8(a)は探索状態等価性を適用した場合である．$x13$ sa1 に対する判定木において，節点6で得られた評価フロンティア $\{(x9,1),(x12,1),(x14,D)\}$ は，$x5$ sa1 に対する判定木の節点6で得られた評価フロンティアと等価になる．従って，$x5$ sa1 に対する判定木の節点6以降の割当てを用いて $x13$ sa1 に対するテスト生成をすることができる．結果として得られるテストパターンは $(x1,x2,x3,x4,x5,x6)=(1,1,1,0,1,1)$ である．図8(b)は探索状態被覆性を用いた場合である．$x13$ sa1 に対する判定木において，節点2で得られた評価フロンティア $\{(x14,D)\}$ は，$x5$ sa1 に対する判定木の節点6で得られた評価フロンティア $\{(x9,1),(x12,1),(x14,D)\}$ を被覆している．評価フロンティアは両方とも活性化しており，$x5$ sa1 に対する判定木の節点6はテストパターンまでの経路をもっている．従って，定理2(b)を用いて $x13$ sa1 に対するテスト生成をすることができる．結果として得られるテストパターンは $(x1,x2,x3,x4,x5,x6)=(1,1,1,0,1,1)$ である．探索状態被覆性を用いた場合，等価性を用いる場合と比較して，更に4ステップ分の探索空間を削減することができる． □

探索履歴の利用に際し，探索状態被覆性を用いることはできても，探索状態等価性は用いることができない場合がある．その例を以下に示す．

[例5] 図9(a)に示す回路について，信号線 $x11$ 上の1縮退故障(以下 $x11$ sa1 と記す)および $x4$ 上の0縮退故障(以下 $x4$ sa0)に対するテスト生成について考える．ここで，もとになるテスト生成アルゴリズムとしてPODEMを用いることにする．

図9(b), (c)はテストパターン探索の過程を2分決定木で表現したものである．図9(b)はPODEMのみを用いてテストパターン探索を行った場合である．$x11$ sa1 では五つの節点，$x4$ sa0 では四つの節点を通過した後，テスト生成に成功している．図9(c)はPODEMに加えて，探索状態被覆性を適用した場合である．$x4$ sa0 に対応する探索木において，節点2で得られた評価フロンティア $\{(x13,D')\}$ は，$x11$ sa1 に対する判定木の節点3で得られた評価フロンティア $\{(x7,1),(x13,D)\}$ を被覆する．$x11$ sa1 の節点3はテストパターンまでの経路をもつので，定理2(b)を用いて $x4$ sa0 のテスト生成を行うことができる．得られるテストパターンは，$(x1,x2,x3,x4,x5,x6)=(0,X,0,1,1,0)$ である．

探索状態被覆性を適用した場合，PODEMに対して3ステップ分の探索空間が削減できる．一方，探索状態等価性を適用しようとしても，図9(b)に見られるように等価な評価フロンティアは存在しない．従ってこの例の場合，探索状態等価性を用いた探索空間削減は不可能である．

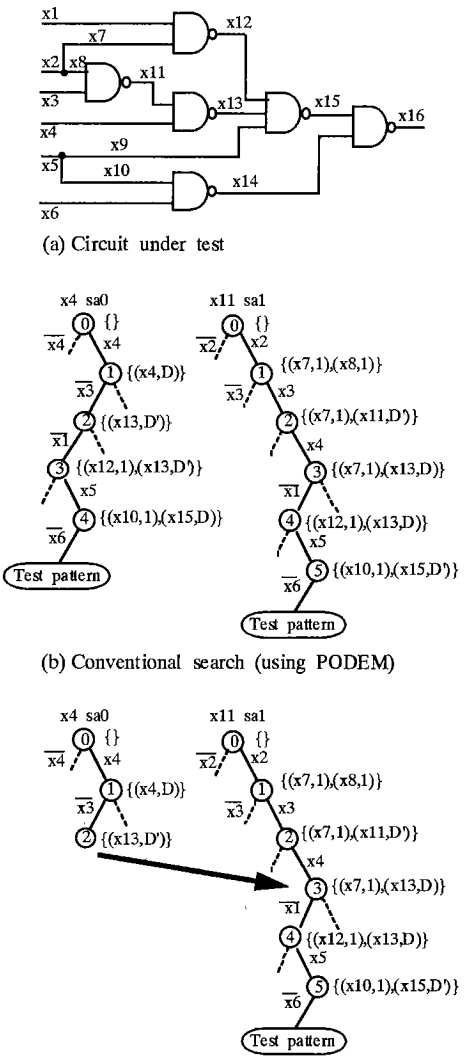


図9 探索状態被覆性を用いた探索空間の削減
Fig. 9 Search space pruning using search state dominance.

4. DST アルゴリズム

前節で定義した探索状態被覆性に基づく DST (Dominant STate hashing) アルゴリズムを述べる．DST アルゴリズムは EST アルゴリズムと同様に，通常のテスト生成における含意操作後の探索状態に対応する評価フロンティアに対して適用するアルゴリズムであるので，既存のテスト生成アルゴリズムに付加して用いられる．

図 10 に DST アルゴリズムのフローチャートを示す．E_c はその時点での探索状態に対応する評価フロンティアを示している．また，E_s は E_c をハッシュしてスタックから取り出された，既に探索済みの評価フロンティアの一つを表す．

履歴として記録する情報は，対象とする故障 (信号線とその種類)，その探索結果 (解をもつか，否か)，解をもつ場合はそのときの外部入力の値である．

DST アルゴリズムはテスト生成アルゴリズムの含意操作終了後開始され，処理結果は以下の三つの場合が考えられる．

(1) テストパターンが生成される．

(2) バックトラックが発生する．

(3) もとになるテスト生成アルゴリズムに戻り，テスト生成処理が継続される．

図 10 に示す DST アルゴリズムにおいて，初めに現在の評価フロンティア E_c を求める．続いて，得られた評価フロンティア E_c をハッシュし，条件を満たす評価フロンティアをすべて取り出した後，スタックに入れる．その後スタックから一つずつ評価フロンティアを抽出し，現在の評価フロンティア E_c との被覆関係を調べ，定理 1，2 を適用する処理に入る．スタックが空になった場合は，もとになるテスト生成アルゴリズムの処理に戻る (継続)．

図に示すように E_c，E_s がともに同じ目標故障についての評価フロンティアでありかつ，E_s が E_c を被覆している場合には定理 1 が適用される．また，E_c，E_s が別の目標故障に対する活性化された評価フロンティアであり，E_s が E_c を被覆している場合には，定理 2 (a) が適用される．どちらの場合においても，バックトラックが発生する．

E_c，E_s がそれぞれ別の目標故障に対する活性化された評価フロンティアであり，E_c が E_s を被覆している場合には，定理 2 (b) を適用してテストパターンを生成する．

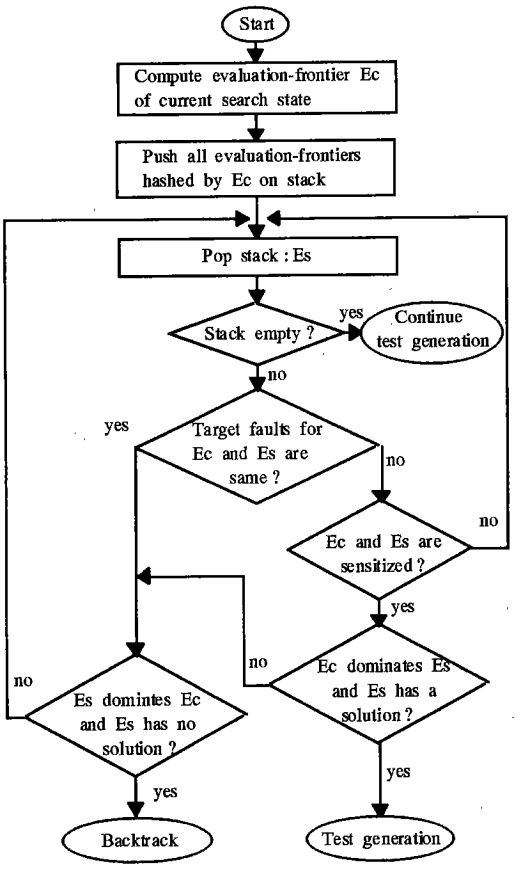


図 10 DST アルゴリズム
Fig. 10 DST algorithm.

5. 実験結果

前節で提案した DST アルゴリズムを，C 言語を用いて SUN4/330 ワークステーション上に構築した．実験は ISCAS '85 ベンチマーク回路に対して，PODEM，PODEM に DST アルゴリズムのみを付加したものをそれぞれ用いてテスト生成を行った．バックトラックリミットはともに 1,000 である．今回の実験では，故障シミュレーションや探索履歴の利用以外の発見的手法は一切用いていない．

表 1 に探索状態被覆性を用いた探索空間削減の効果を示す．表中において，#Backtracks はバックトラックの回数を表し，#Implications は含意操作の回数，すなわち探索木における節点の数を表す．Δ は探索状態被覆性を用いることによって，もとになる PODEM に対してバックトラック数，含意操作数がどれだけ減少したかを示すものである．バックトラック数は C880，

表1 探索状態被覆性による探索空間削減の効果

回路名	#Backtracks			#Implications		
	PODEM	PODEM +DST	Δ	PODEM	PODEM +DST	Δ
C432	44020	44017	-3	50313	48944	-1369
C499	8688	8652	-36	35958	26398	-9560
C880	1	1	0	8525	7037	-1488
C1355	8482	8335	-147	64676	38081	-26595
C1908	9118	9038	-80	35305	27216	-8089
C2670	152801	71864	-80937	193798	96390	-97408
C3540	210508	208769	-1739	249083	241927	-7156
C5315	16422	15725	-697	82349	61167	-21182
C6288	278941	278941	0	471419	429176	-42243
C7552	160329	154693	-5636	323505	292718	-30787

表2 探索状態等価性および探索状態被覆性の適用回数の比較

回路名	#Hash Tests			#Hash Backtracks		
	等価性	被覆性	Δ	等価性	被覆性	Δ
C432	68	145	77	1983	1988	5
C499	188	277	89	0	0	0
C880	196	278	82	0	0	0
C1355	600	785	185	0	0	0
C1908	759	842	83	25	25	0
C2670	872	1221	349	3988	9169	5181
C3540	678	916	238	3477	4543	1066
C5315	1360	2048	688	67	100	33
C6288	2473	2949	476	0	0	0
C7552	1031	1406	375	509	523	14

C6288を除くすべての回路について減少している．また，含意操作の回数はすべての回路で減少している．特にC1355，C2670の二つの回路については含意操作の回数，すなわち探索空間を40％～50％削減することに成功しており，C499，C1908，C5315についても25％前後の探索空間を削減している．

表2は，PODEM＋DSTを用いたテスト生成における，探索状態等価性および探索状態被覆性の適用回数をそれぞれ示したものである．#Hash Testsは探索履歴を用いてテストパターン生成を行った回数，#Hash Backtracksは探索履歴を用いることによって発生したバックトラックの回数である．Δは探索状態等価性の適用回数に対して，探索状態被覆性がどれだけ多く適用されたかを示すものである．#Hash Testsでは，すべての回路において探索状態被覆性の適用回数が探索状態等価性の適用回数を上回っている．また，#Hash Backtracksについても，C432，C2670，C3540，C5315，C7552の五つの回路で探索状態被覆性が探索状態等価性に比べ，多く適用されている．これは探索状態被覆性は探索状態等価性と比較して，より多くの探索履歴が利用可能であることを示している．従って，探索履歴を利用した探索空間削減において，探索状態被覆性は探索状態等価性よりも効果的に探索空間を削減することがわかる．

表3 打切り故障数，故障検出率，CPUタイムの比較

回路名	故障数	打ち切り故障数		故障検出率（％）		CPU time（秒）	
		PODEM	PODEM +DST	PODEM	PODEM +DST	PODEM	PODEM +DST
C432	524	42	42	91.18	91.18	132.7	416.2
C499	758	8	8	98.94	98.94	283.1	384.5
C880	942	0	0	100.00	100.00	40.9	38.6
C1355	1574	8	8	99.49	99.49	466.4	469.3
C1908	1879	9	9	99.20	99.20	339.7	483.0
C2670	2747	124	42	93.37	95.49	2700.0	5232.9
C3540	3428	169	166	92.91	93.00	4185.2	11660.7
C5315	5350	7	7	98.84	98.84	2763.8	2428.4
C6288	7744	208	208	96.90	96.90	18629.1	86049.0
C7552	7550	145	142	97.30	97.30	15569.7	127697.4

表3は打切り故障数，故障検出率，CPUタイムについて，DSTアルゴリズムを用いて探索空間削減を行った場合と行わない場合を比較したものである．C2670，C3540において，もとになるPODEMに対して故障検出率の向上が見られ，C2670，C3540，C7552において打切り故障数が減少している．特に，C2670では打切り故障数は35％以下に減少している．本実験では評価フロンティア蓄積のためのメモリ制限によりバックトラックリミット1,000という限られた探索空間内でテストパターンを探索しているために打切り故障が残っている．これはバックトラックリミットを十分に大きく設定し，一意活性化[5]や学習[6]等の探索空間削減のための手法を付加することで0にすることが可能であると考えられるが，今回の実験では探索空間削減のための処理が一切なされていない探索空間を対象に実験を行うために，これらの手法は用いなかった．CPUタイムについては，探索履歴を用いる場合ほとんどの回路について，もとになるPODEMよりも多くの時間を必要とする．これは，DSTアルゴリズム実行時にハッシュの衝突が多発することによるもので，より高速かつ効率的なハッシュ関数を用いることで，ある程度改善が期待できる．

6. むすび

本論文では，探索状態被覆性という新しい概念に基づいた，DSTアルゴリズムを提案した．探索状態被覆性は，GiraldiとBushnellの提案したESTアルゴリズムにおける探索状態等価性の概念を拡張したものであ

る．実験結果は，探索状態被覆性は探索状態等価性と比較して更に探索空間の削減が可能であることを示している．しかし，探索状態被覆性を用いる場合，ハッシュ時の衝突が多く，ハッシュ処理に時間がかかってしまう問題がある．また，評価フロンティアの蓄積には大量のメモリを必要とするので，回路がより大規模化した際に記憶容量によって探索履歴の適用範囲が制限されてしまうことが考えられる．従って，より効率的なハッシュ法の検討および評価フロンティア蓄積の省容量化などが今後の課題である．

文　献

(1) Fujiwara H.: "Logic Testing and Design for Testability", the MIT Press (1985).

(2) Brglez F. and Fujiwara H.: "A Neutral Netlist of 10 Combinational Benchmark Circuits", Proc. IEEE ISCAS; Special Session on ATPG and Fault Simulation, pp. 151-158 (June 1985).

(3) Roth J. P., Bouricius W. G. and Schneider P. R.: "Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits", IEEE Trans. Electronic Computers, **EC-16**, 10, pp. 567-580 (Oct. 1967).

(4) Goel P.: "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", IEEE Trans. Comput., **C-30**, 3, pp. 215-222 (March 1981).

(5) Fujiwara H. and Shimono T.: "On the Acceleration of Test Generation Algorithms", IEEE Trans. Comput., **C-32**, 12, pp. 1137-1144 (Dec. 1983).

(6) Schulz M. and Auth E.: "Improved Deterministic Test Pattern Generation with Applications to Redundancy Identification", IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst., 8, 7, pp. 811-816 (July 1989).

(7) Larrabee T.: "Efficient Generation of Test Patterns using Boolean Difference", Prof. IEEE Int. Test Conf., pp. 795-801 (Aug. 1989).

(8) Giraldi J. and Bushnell M. L.: "EST: The New Frontier in Automatic Test-Pattern Generation", Proc. 27th-DAC, pp. 667-672 (June 1990).

(9) Giraldi J. and Bushnell M. L.: "Search State Equivalence for Redundancy Identification and Test Generation", Prof. IEEE Int. Test Conf., pp. 184-193 (Aug. 1991).

(10) 浅野純也，藤野貴之，藤原秀雄："履歴を利用したテスト生成の一手法"，信学技報，**FTS91**-6 (1991-04).

(11) Fujino T. and Fujiwara H.: "An Efficient Test Generation Algorithm Based on Search State Dominance", FTCS-22, pp. 246-253 (Sept. 1992).

(平成4年9月14日受付，12月24日再受付)

藤野　貴之

平1明大・工・電子卒．平3同大大学院博士前期課程了．現在，同大大学院博士後期課程在学中．ニューラルネット，テスト生成アルゴリズムに関する研究に従事．

藤原　秀雄

昭44阪大・工・電子卒．昭49同大大学院博士課程了．同年同大電子工学教室助手．昭60明治大・工・助教授．現在，同大理工学部教授(情報科学科)．昭56ウォータルー大客員助教授．昭59マッギル大客員準教授．コンピュータの設計とテスト，テスト容易化設計，テスト生成，並列処理，ニューラルネット，計算複雑度に関する研究に従事．著書「論理数学の基礎」(共著)，「ディジタル回路の故障診断」(共著)，「Logic Testing and Design for Testability」，「コンピュータの設計とテスト」．情報処理学会，日本教育工学会各会員．IEEE Fellow．