特　集

モデル列挙とモデル計数

Model Enumeration and Model Counting

長谷川 隆三　九州大学大学院システム情報科学研究院
Ryuzo Hasegawa　Graduate School of Information Science and Electrical Engineering, Kyushu University
hasegawa@ar.is.kyushu-u.ac.jp, http://zircon.is.kyushu-u.ac.jp/~hasegawa/

藤田 博　（同　　上）
Hiroshi Fujita
fujita@is.kyushu-u.ac.jp, http://zircon.is.kyushu-u.ac.jp/~fujita/

越村 三幸　（同　　上）
Miyuki Koshimura
koshi@inf.kyushu-u.ac.jp, http://zircon.is.kyushu-u.ac.jp/~koshimura/

keywords: model generation, minimal models, model enumeration, model counting

1. はじめに

SAT 問題および SAT ソルバーにおいては，通常，反駁法に基づく一階述語論理の定理証明と同様に，与式が充足可能か否かの判定を下すことに主たる興味がある．実際，Chaff [Moskewicz 01] や MiniSat [Eén 05] に代表される DPLL [Davis 60, Davis 62] に基づく SAT ソルバーは元来，命題の反駁証明器（単解探索）であり，全解探索目的には作られていない．これに対し，MGTP [長谷川 08] のようなモデル生成法に基づく定理証明器は反駁証明にも用いられるが，本質的にモデル発見器である．DPLL とは異なり，MGTP は EPR(Effectively Propositional) 論理のための一階述語表現ができる．これらはいずれも完全型 SAT ソルバーであるが，Walksat [Selman 93] に代表される確率統計的手法に基づく不完全型 SAT ソルバーもある．本稿で取上げるソルバーと用途，技術の相互関係を図 1 に示す．

用途に視点を移すと，与式を充足する解釈／モデルの具体例を知りたいことがある．さらには，そのような解釈／モデルが複数ある場合，それら全てを求めたいこともある．これを**モデル列挙** (model enumeration)(AllSAT) という．モデル列挙技術の応用例には，命題式の CNF 変換，限量子除去，制限無しモデル検査などがある [Biere 99, McMillan 02]．一方，与式を充足するモデルそのものよりもモデルの総数にのみ興味がある応用もある．これを**モデル計数** (model counting)(#SAT) という．モデル計数技術については，プランニング，ベイズネット推論，不完全情報を用いた推論，記号モデル検査，CNF → DNNF 変換，制約充足問題の解の個数の計算など，多数の実際的応用がある [Grove 94, Roth 96, Kauz 96, Darwiche 04]．

単位伝播と矛盾節学習は，完全型 SAT ソルバーの探索空間を刈込むための必須技術であり，DPLL のみならず，モデル生成型にも採り入れられている．モデル列挙では，重複は厭わず，全モデルを被覆する最小個数の部分割り

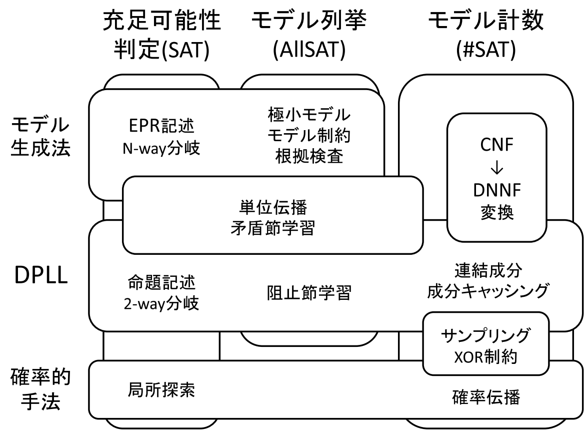


図 1　本解説が焦点を当てる技術マップ

当てを得ることが目的となる．そこで，矛盾節学習に加えて阻止節学習が導入された [McMillan 02]．モデル生成法においては，上記目的のため，極小モデルの概念が導入され，モデル制約 [Bry 96] や根拠検査 [Niemelä 96] が開発された．一方，モデル計数の場合，計数精度，即ち，重複無しにモデルを数え上げることが重要になる．学習機構としては，専ら，矛盾節学習のみが使用されている．モデル計数のための DPLL ベースのアルゴリズム CDP が [Birnbaum 99] で初めて提示された．その後，CNF 式中の連結成分の存在を利用することにより，CDP のアルゴリズムを改善した DDP が提示され，Relsat 上に実装された [Bayardo 00]．さらに，#SAT の有望な方式として成分キャッシングが提案され，Cachet 上に実装された [Sang 04]．一方，確率統計手法に基づく#SAT の研究においては，サンプリング，XOR 制約，確率伝播などの技法が開発されている．

本解説では，モデル列挙とモデル計数の主要技術を解説する．

2. 準　　備

連言標準形（CNF）の論理式 F は，変数の有限集合 $\mathcal{V}$ 上で定義される，節の連言からなる．ただし，各節 ω はリテラルの選言であり，各リテラル l は変数 x_i か，その否定 $\neg x_i$ である．以後，対象 X の否定（補）を $\neg X$ あるいは $\overline{X}$ と表す．CNF 式 F は節の集合，節 ω はリテラルの集合とみなす．

割り当て π は $\mathcal{V}$ から $\{0,u,1\}$ への関数である．ただし，0 は偽，1 は真，u は未指定値を表す．もし $\forall x_i \in \mathcal{V}\ \pi(x_i) \in \{0,1\}$ ならば，割り当て π は完全であるといい，さもなければ部分的という．割り当て π は対 (x_i,v_i) の集合，$\pi = \{(x_1,v_1),\ldots,(x_n,v_n)\}$ とみなせる．ただし，$v_i = \pi(x_i) \in \{0,u,1\}$．$(x_i,u)$ を未指定変数割り当てといい，$(x_i,0),(x_i,1)$ を指定変数割り当てという．また，決定変数に対する割り当てを決定変数割り当て，単位伝播により含意される変数に対する割り当てを含意変数割り当てという．

割り当て π の下での式 φ の値を $\varphi|_\pi$ と表す．$\varphi|_\pi = 1$ となる π を充足割り当てといい，$\varphi|_\pi = 0$ となる π を非充足割り当てという．π' が π 中の指定変数割り当てを全て含めば，π' は π に被覆されるという．$\forall x_i \in \mathcal{V}\ \pi(x_i) = \pi'(x_i) \vee \pi(x_i) = u \vee \pi'(x_i) = u$ のとき，割り当て π と π' は交叉するという．式 φ のモデルは，$\varphi|_\pi = 1$ なる完全割り当て π のことである．

以後、モデル生成法のように 1 階述語を扱う場合は、命題文字 (propositional letter) を “アトム” と称し，DPLL のように命題のみを扱う場合は，“変数” と称する．

任意の節 $C = L_1 \vee \ldots \vee L_k$ は，C 中の負リテラルを $\neg A_1,\ldots,\neg A_m$，正リテラルを $B_1,\ldots,B_n$ とすると，含意式 $A_1 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \ldots \vee B_n$ で表現できる．“$\rightarrow$” の左側を前件，右側を後件という．$m=0$ のとき正節，$n=0$ のとき負節，$m>0$ かつ $n>0$ のとき混合節という．アトムの集合 M について，

$\forall i(1 \leq i \leq m) A_i \in M \wedge \forall j(1 \leq j \leq n) B_j \notin M$

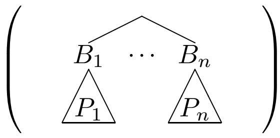
が成り立つ節を M の**違反節** (violated clause) という．与えられた節集合 S に M の違反節が存在しないとき，次のように解釈するモデルで S は充足される：「M に現れるアトムは真であり，現れないアトムは偽である.」

3. モ デ ル 列 挙

本節では，モデル生成に基づくモデル列挙と DPLL に基づくモデル列挙の概説を行う．

3·1 モデル生成法

図 2 にモデル生成法による証明手続きを示す．関数 MG は，真が割り当てられたとみなすアトムの集合 Mc（モデル候補）と節集合 S を受け取り，S の（部分）証明木を返す．Mc は空集合 $\emptyset$ に初期設定される．モデル棄却された枝は閉じた (closed) といわれ，その葉は $\perp$ でラベル付けされる．一方，閉じない枝は開放枝 (open branch) といわれ，葉が $\top$ とラベル付けされるか，枝が無限に伸びていくかのいずれかになる．開放枝の Mc には S の違反節がないので，Mc は S のモデルと解釈できる．このように開放枝が一つでもあれば，S は充足可能である（モデル生成法の完全性）．一方，全ての枝が閉じられた場合，S は充足不能である（モデル生成法の健全性）．

関数 $\mathrm{MGTP}(S)$;　/* S：節集合 */
　$(\mathrm{MG}(\emptyset,S))$ を返す.
関数 $\mathrm{MG}(Mc,S)$;　/* Mc：モデル候補 */
1. **モデル棄却** 負節 $(A_1 \wedge \ldots \wedge A_m \rightarrow) \in S$ が，Mc の違反節の場合：$(\overset{|}{\perp})$ を返す.
2. **モデル拡張**　混合節もしくは正節 $(A_1 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \ldots \vee B_n) \in S$ が，Mc の違反節の場合：全ての i $(1 \leq i \leq n)$ について，
$P_i = \mathrm{MG}(Mc \cup \{B_i\},S)$ とおいて，


を返す.
3. **モデル獲得** 1. も 2. も適用できない場合：
$(\overset{|}{\top})$ を返す.

図 2　モデル生成手続き

3·2 極小モデル生成

命題式のモデルは，そのモデルで真と解釈されるアトムの集合で表現することができる．例えば，a を真，b を偽，c を真，と解釈するモデルはアトム集合 $\{a,c\}$ で表現できる．モデルをこのように表現すると，集合の包含関係を用いて，モデル間に大小関係を自然に導入できる．例えば，モデル $\{a,c\}$ は，モデル $\{a,b,c\}$ より小さい．本節では，モデル間にこのような大小関係を導入した時の，極小モデルに焦点をあてる．

【定義 1】 M_1，M_2 をアトム集合とする．このとき，M_1 が M_2 より小さいとは，M_1 が M_2 の真部分集合であることをいう．

【定義 2】（極小モデル） φ を命題式，M を φ のモデルとする．このとき，M が φ の極小モデルである，とは，M より小さい φ のモデルが存在しないことをいう．

〔例 1〕 φ を命題式とする．そして，φ には三つのモデル $M_1 = \{p_1,p_2,p_3\}$，$M_2 = \{p_1,p_2\}$，$M_3 = \{p_3\}$ があるものとする．このとき，M_2 と M_3 は極小モデルだが，M_1 は極小でない．

前節で述べたモデル生成法は，全ての極小モデルを列挙できることが知られている [Bry 96].

［定理 1］（極小モデル完全性） S を節集合，M をその極小モデルとする．このとき，モデル生成手続き MGTP(S) で得られる証明木の開放枝に，その対応するモデル Mc が M と等しいものがある．

このように，モデル生成法は，極小モデルを全て列挙することができる．しかし，一般的に極小でないモデルも列挙してしまう．本節では以降，全ての極小モデルを列挙し，かつ極小モデルのみを列挙する手法を紹介する．

§ 1　Bry の極小モデル生成

Bry らは，モデルを非降順に列挙するために，図 2 のモデル拡張を次のように変更した [Bry 96].

［相補分割によるモデル拡張］ 混合節もしくは正節 $(A_1 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \ldots \vee B_n) \in S$ が，Mc の違反節の場合：全ての i $(1 \leq i \leq n)$ について，
$P_i = MG(Mc \cup \{B_i, \neg B_{i+1}, \ldots, \neg B_n\}, S)$ とおいて，

$$\left(\begin{array}{ccccc} B_1 & \cdots & B_i & \cdots & B_n \\ \neg B_2 & & \neg B_{i+1} & & P_n \\ \vdots & & \vdots & & \\ \neg B_n & & \neg B_n & & \\ P_1 & & P_i & & \end{array}\right)$$ を返す．

さらに，各アトム p について，矛盾律を表す負節 $p \wedge \neg p \rightarrow$ を節集合 S に加えると次の定理が成り立つ．

［定理 2］ S を節集合とする．そして，相補分割によるモデル拡張を行うモデル生成手続き MGTP(S) で得られる証明木中のモデルを左から順に $M_1, \ldots, M_k$ とおく．このとき，$\forall i \forall j (1 \leq i < j \leq k)(M_i \not\supseteq M_j)$ が成り立つ．

この定理は，証明木中，左のモデルが右のモデルより大きくも等しくもない，ことを示している．したがって，証明木中のあるモデル M_i が極小であることを示すには，それより左側に現れるモデルより大きくない，ことを示せばよい．つまり，$\forall j (1 \leq j < i)(M_j \not\subset M_i)$ であればよい．なお，証明木中の最左のモデルは，常に極小である，ことに注意されたい．

Bry らは，これらの性質を利用して，証明木を左から右へ深さ優先に構築して極小モデルを過不足なく列挙する手法を提案した．

左から右へ構築するので，最初に見つかったモデル M_1 は無条件に極小となる．そして，次の極小モデルを探索する前に，節集合に負節 $A^1_1 \wedge \ldots \wedge A^1_{m_1} \rightarrow$ を加える．ここで，$\{A^1_1, \ldots, A^1_{m_1}\}$ は M_1 中の正リテラルの集合である．この負節を**モデル制約** (model constraint) と呼ぶ．これにより，以降の探索では M_1 より大きいか等しいモデルは，モデル棄却される．したがって，次に見つかるモデル M_2 は極小である，ことが保証される．

以降，モデル M_i が見つかるたびに，負節 $A^i_1 \wedge \ldots \wedge A^i_{m_i} \rightarrow$ を節集合に加えていく．これにより得られるモデルは全て極小となる．なお，Bry らは，このように負節を加えていく探索を**制約下探索** (constrained search) と呼んでいる．また，Prolog による簡潔な実装も示している．

§ 2　Niemelä の極小モデル生成

Niemelä は，モデルが極小であるための必要十分条件を示した [Niemelä 96].

［定理 3］ φ を命題式，M を φ のモデルとする．また，$M = \{a_1, \ldots, a_m\}$，$\overline{M} = \{b_1, \ldots, b_n\}$ とする．このとき，M が φ の極小モデルであれば，$\Phi(= \varphi \wedge \neg(a_1 \wedge a_2 \wedge \ldots \wedge a_m) \wedge \neg b_1 \wedge \neg b_2 \wedge \ldots \wedge \neg b_n)$ は充足不能であり，逆もまた正しい．

Niemelä は，Φ を充足不能とするような M を根拠がある (grounded) と呼んでいる．このことから，Φ の充足判定を**根拠検査** (groundedness test) と呼ぶ．Niemelä は，相補分割によるモデル拡張とほぼ同等の拡張規則を取り入れたモデル生成手続きでモデルを列挙し，得られたモデルの根拠検査を行うことにより，極小モデルを過不足なく列挙する手法を提案した．

§ 3　長谷川らの改良

Bry や Niemelä の手法は次のような欠点をもつ．すなわち，証明木を解析すれば簡単に極小性が確認できるようなモデルに対して，不要な極小性検査（制約下探索あるいは根拠検査）をおこなってしまうこと，および，非極小モデルを導くような冗長な探索枝を完全には刈りきれないことである．

このような問題を解決するため，長谷川らは相補分割を強化した分岐補題を導入し冗長な探索枝を刈り込み，証明の依存性を解析することで不要な極小性検査を削減する手法を提案した [Hasegawa 00, 長谷川 01, Koshimura 03]．そして，Bry らの手法に比べ，顕著な速度向上を確認している．なお，相補分割と分岐補題の使用は，因子化（factorization）[Letz 94] の適用の一種とみなせる．

§ 4　アトム集合に関する極小モデル

モデルの極小性については，全てのアトム集合に関してではなく，ある特定のアトム集合に関しての極小性が有用なこともある．本節では，与えられたアトム集合に関して極小なモデル生成について考察する．

【定義 3】 P，M_1，M_2 をアトム集合とする．このとき，M_1 が M_2 より P に関して小さい，とは，$M_1 \cap P$ が $M_2 \cap P$ の真部分集合であることをいう．

〔例 2〕 $M_1 = \{p_1, p_2, p_3, a\}$，$M_2 = \{p_1, p_3, b, c, e, f\}$，$P = \{p_1, p_2, p_3\}$ とすると，M_2 は，P に関して M_1 より小さい．

【定義 4】（P に関して極小なモデル） φ を命題式，P をアトム集合，M を φ のモデルとする．このとき，M が P に関して φ の極小モデルである，とは，P に関して M より小さい φ のモデルが存在しないことをいう．

〔例 3〕 φ を命題式，$P = \{p_1, p_2, p_3\}$ とする．そして，φ には三つのモデル $M_1 = \{p_1, p_2, p_3, a\}$，$M_2 = \{p_1, p_2, b\}$，$M_3 = \{p_3, b\}$ があるものとする．このとき，M_2 と M_3 は P に関して極小モデルだが，M_1 は極小でない．

次の定理は，定理 3 の自然な拡張になっている．

［定理 4］ φ を命題式，P をアトム集合，M を φ のモデルとする．また，$M \cap P = \{a_1, \ldots, a_m\}$，$\overline{M} \cap P = \{b_1, \ldots, b_n\}$ とする．このとき，M が φ に関して φ の極小モデルであれば，$\Phi(= \varphi \wedge \neg(a_1 \wedge a_2 \wedge \ldots \wedge a_m) \wedge \neg b_1 \wedge \neg b_2 \wedge \ldots \wedge \neg b_n\ )$ は充足不能であり，逆もまた正しい．

〔例 4〕 φ を命題式，$P = \{p_1, p_2, p_3, p_4\}$ とする．このとき，φ のモデル $\{p_1, p_4, c, d\}$ が P に関して極小モデルであれば，$\varphi \wedge \neg(p_1 \wedge p_4) \wedge \neg p_2 \wedge \neg p_3$ は充足不能であり，逆も成り立つ．

越村らは，与えられたアトム集合に関して極小なモデル生成を既存の SAT ソルバーを利用して実装する手法を示した [越村 08]．この手法では，SAT ソルバーを利用して列挙したモデルから，定理 4 に基づく根拠検査により，非極小モデル除外する．また，制約下探索をおこなうことにより，SAT ソルバーが同じモデルを何度も列挙するのを避けている．

なお，定義 4 は，極小化対象述語を P としたときの極小限定 (circumscription) と同じである．極小限定などの非単調推論の論理は，解集合プログラミングへ受け継がれており，解集合ソルバが利用できるようになっている．解集合ソルバを利用して，極小モデルを求めることができる．解集合プログラミングについては，井上らの解説 [井上 08] を参照されたい．

3·3　DPLL に基づくモデル列挙

SAT ソルバーでは，非充足割り当てから**矛盾節** (conflict clause) を作り，冗長な探索を回避するのが効果的であるが，充足割り当てからも類似の節，**阻止節** (blocking clause) を作れば，モデルの重複計算を回避し，簡潔なモデル集合表現を得ることができる．これは，極小モデル生成の冗長性を回避するモデル制約に似ている．本節では，DPLL 型 SAT ソルバーにおけるモデル列挙の主要技術として，阻止節を概説する．

§ 1　阻止節

与えられた CNF 式を F，現在の充足割り当てを π とする．π が被覆するモデル集合を除外するため，π の補をとり，それを初期阻止節 c とする．阻止節は，π の下で偽であり，$F \wedge c$ の論理的帰結である，という性質を満たす．阻止節は，矛盾節と類似の方法で求められる．充足割り当てが見つかる度に，まず，上述の初期阻止節を生成する．これを阻止節としてもよいが，さらに簡約された阻止節は，これと元の CNF 式との融合により生成できる．

阻止節を用いると，任意の命題式 φ の CNF 変換が可能になる．φ の CNF 式 $\mathrm{CNF}(\varphi)$ は，その否定 $\neg\varphi$ の積和形（モデル集合）を求め，その補をとれば求まる．例えば，$\varphi = a \vee (b \wedge c)$ とすると，$\neg\varphi = (\neg a \wedge \neg b) \vee (\neg a \wedge \neg c)$．$\therefore \mathrm{CNF}(\varphi) = \neg\neg\varphi = (a \vee b) \wedge (a \vee c)$．

$\neg\varphi$ の積和形を得るために，$f = \mathrm{CNF}(\varphi) \wedge \neg l_\varphi \wedge \chi$ に対し，阻止節学習機構を組込んだ DPLL アルゴリズムを適用する．ただし，$l_\varphi \leftrightarrow \mathrm{CNF}(\varphi)$，$\chi$ は阻止節の集合とする．阻止節の定義より，$\varphi \models \chi$ である．アルゴリズムが停止した際，$f = 0$ ならば，$\chi \models l_\varphi$ である．よって，このとき，χ は φ と同値な CNF 式を与える．この CNF 変換アルゴリズムは，順に，限量子除去，制限無しモデル検査のアルゴリズムへと発展させることができる [McMillan 02]．

§ 2　阻止節の最小化

最も簡単な阻止節は充足割り当ての補をとるだけのものであるが，より探索空間削減効果を上げるには，ブロッキング節の最小化が課題になる．例えば，前節の CNF 変換 [McMillan 02] では，各部分式を表す補助変数が導入されるが，阻止節はこれらを除いた元の式中の変数のみからなるものに限定されている．

Jin らは，矛盾節学習と組合わせることにより，阻止節を最小化する方法を提示した [Jin 05]．探索空間は非充足割り当てを表す UNSAT と充足割当を表す SAT の空間に分けられるが，従来の矛盾節と阻止節はそれぞれ，UNSAT と SAT の刈込にしか適用できない．そこで，充足割り当てから得られた初期阻止節を UNSAT 側まで被覆するように最小化する．主要アイデアは，更新中の式 F（＝入力式＋阻止節＋矛盾節）に対して，初期阻止節 β を矛盾節とみなし，β に関するコンフリクト解析を行い，矛盾節 γ を生成することである．融合された γ は，充足可能点（充足割り当て）と充足不能点（非充足割り当て）の両方を包含するので，より高い刈込効果が得られる．$F \wedge \beta \models \gamma$ であり，γ は β に含まれない充足割り当てを阻害 (block) しないことに注意．

4.　モデル計数 (#SAT)

命題モデル計数 (#SAT) は，与えられた命題式 φ のモデルの個数 $\#\varphi$ を求める問題である．#SAT はベイズネット推論と関連が深く，応用上も重要である．

SAT と#SAT は，それぞれ代表的な NP 完全問題，#P 完全問題である．一般に#SAT は SAT より難しい．特に，2SAT は多項式時間で易しいが，そのモデル計数は依然 #P 完全問題で難しい [Valiant 79]．

4·1　正確なモデル計数

正確なモデル計数を与える#SAT ソルバーのほとんどが，MiniSat などの完全探索型 SAT ソルバーと同様，DPLL 方式に基づいている．その代表例としては，CDP, Relsat, Cachet, sharpSAT, c2d などがある．

§ 1　DPLL 型の素朴な#SAT

Birnbaum らは，命題式の完全充足割り当て（モデル）の個数を正確に計算する素朴な#SAT アルゴリズムを提示した [Birnbaum 99]．本アルゴリズムは DPLL 手続に

関数 $\mathrm{CDP}(F,\ n)$
1. もし F が空ならば，2^n を返す．
2. もし F が空節を含めば，0 を返す．
3. もし F が単位節 $\{l\}$ を含めば，
   $F_1 = \{C - \{\bar{l}\} \mid C \in F, l \notin C\}$.
   $\mathrm{CDP}(F_1, n-1)$ を返す．
4. F 中の変数 x を選ぶ．
   $F_1 = \{C - \{\neg x\} \mid C \in F, x \notin C\}$.
   $F_2 = \{C - \{x\} \mid C \in F, \neg x \notin C\}$.
   $\mathrm{CDP}(F_1, n-1) + \mathrm{CDP}(F_2, n-1)$ を返す．

図 3 素朴な DPLL 型#SAT (CDP)

手続 $\#\mathrm{DPLLcache}(F)$
1. F 中のキャッシュ済成分を除去する．
2. もし F が空ならば，帰る．
3. ある成分 $\phi \in F$ 中の変数 v を選ぶ．
   $\phi|_{\overline{v}}$ を成分に分解し，結果を F^- とする．
   $\#\mathrm{DPLLcache}((F - \{\phi\}) \cup F^-)$.
   $\phi|_v$ を成分に分解し，結果を F^+ とする．
   $\#\mathrm{DPLLcache}((F - \{\phi\}) \cup F^+)$.
   $(\phi,\ \mathrm{GetValue}(F^-) \times \frac{1}{2} + \mathrm{GetValue}(F^+) \times \frac{1}{2})$
   をキャッシュに登録する．

図 4 成分キャッシングつき DPLL 型#SAT

基づいており，CDP(Counting by Davis-Putnam) と呼ばれる．CDP 関数を図 3 に示す．F は命題 CNF 式，n は F 中の変数の個数である．

DPLL の分割規則は，変数 x を選び，F を $(x \wedge F|_x) \vee (\overline{x} \wedge F|_{\overline{x}})$ へシャノン展開する．これにより，F のモデル集合は，互いに疎な，$F_1 = F|_x$ のモデル集合と $F_2 = F|_{\overline{x}}$ のモデル集合に分割される．したがって，x に真（偽）を割り当て，$F_1(F_2)$ に出現しない F の全ての変数には勝手に真または偽を割り当てることにより，$F_1(F_2)$ の各モデルを F のモデルに拡張することができる．F に出現するが，$F_1(F_2)$ には出現しない (x 以外の）変数の数を $n_1(n_2)$ とすると，$\#F = 2^{n_1}\#F_1 + 2^{n_2}\#F_2$ となる．

§2　成分に基づく計数

CNF 式 F が部分式 $F_1 \wedge \ldots \wedge F_k$ に分解されて，どの F_i の対にも共有変数がない場合，$\#F = \#F_1 \times \ldots \times \#F_k$ である．このような上手い分割があれば，#SAT は簡単になる．上記を満たす F_i を F の**成分** (component) と呼ぶ．

これを実現するため，Bayardo らは，前述のモデル計数アルゴリズム CDP を修正し，成分を動的に検出・利用できるようにした版，DDP(Decomposition Davis-Putnam) を提示した [Bayardo 00]．DDP は与えられた CNF 式 F と割り当てに対して単位伝播を行ない，簡約後の F の成分 $F_1, \ldots, F_n$ を識別する．各 F_i の再帰実行の結果得られる $\#F_i$ は最後に掛合され，$\#F$ が得られる．

$F = (a \vee b \vee c) \wedge (p \vee q \vee r) \wedge (\neg p \vee x \vee y)$ を考えよう．簡単な分析により，F は $F_1 = (a \vee b \vee c)$, $F_2 = (p \vee q \vee r) \wedge (\overline{p} \vee x \vee y)$ の 2 成分に分解されることが分かる．

まず，F_1 の解を探す．変数 a を選択し a に真を割り当てれば，F_1 は充足する．未割り当て変数 b, c が残るので，倍率 2^2 をかけて，$\#F_1|_a = 4$ となる．他方，a に偽を割り当てると，$F_1|_{\overline{a}}$ は未充足であるが，次に b を選択し b に真を割り当てれば充足する．未割り当て変数 c が残るので，倍率 2^1 をかけて，$\#F_1|_{\overline{a},b} = 2$ となる．次の $F_1|_{\overline{a},\overline{b}}$ は c を含意して充足するので，$\#F_1|_{\overline{a},\overline{b},c} = 1$ となる．F_1 の各部分問題のモデル集合は互いに疎だから，$\#F_1 = \#F_1|_a + \#F_1|_{\overline{a},b} + \#F_1|_{\overline{a},\overline{b},c} = 4 + 2 + 1 = 7$ と求まる．

次に，F_2 の解を探す．変数 p を選択し p に真を割り当てると，$F_2|_p = (q \vee r)$ を得る．$F_1|_{\overline{a}}$ 以降と同様にして，$\#F_2|_p = 3$ を得る．他方，p に偽を割り当てると，$F_2|_{\overline{p}} = (x \vee y)$ を得て，同様に $\#F_2|_{\overline{p}} = 3$ を得る．$F_2|_p$ と $F_2|_{\overline{p}}$ のモデル集合は互いに疎なので，$\#F_2 = \#F_2|_p + \#F_2|_{\overline{p}} = 3 + 3 = 6$ となる．

最後に，F_1 と F_2 は F の独立成分だったから，$\#F = \#F_1 \times \#F_2 = 7 \times 6 = 42$ と計算される．

§3　成分キャッシング

探索中に記憶しておくと都合のよい情報が，何種類か考えられる．とりわけ，F の成分をキャッシュして，再計算を避けるのが効果的である．Cachet [Sang 04] が図 4 のような手続きを基本として実装された．非充足割り当てに対する学習機構を有する Chaff をベースとしているが，この学習機構と成分キャッシュの組み合わせには微妙な問題があり，素朴に行うとモデル数の下界しか得られない．

一般に，入力 CNF 式 F が $F = A \wedge B$ のように分解され，C を $F \models C$ なる節，即ち，F からの学習節とするとき，ある部分割り当て π が存在して，

(i) $F|_\pi$ は互いに疎な成分 $A|_\pi$ と $B|_\pi$ に分解される．
(ii) $C|_\pi$ は $A|_\pi$ の変数のみを含む．
(iii) $Pr(A|_\pi) \neq Pr(A|_\pi \wedge B|_\pi)$.

となることがある．

例えば，A を $p_0 \vee \neg a_1 \vee p_1$, $p_0 \vee \neg p_2 \vee a_2$, $a_1 \vee \neg a_1 \vee a_3$ の連言，B を $\neg p_1 \vee b_1$, $\neg b_1 \vee b_2$, $\neg b_2 \vee p_2$ の連言とすると，$C = p_0 \vee \neg a_1 \vee a_2$ は，$F = A \wedge B$ の帰結である．

π を $\{(p_0,0),(p_1,1),(p_2,0)\}$ とする．$A|_\pi = (a_1 \vee a_2 \vee a_3)$ と $B|_\pi = (b_1)(\neg b1 \vee b_2)(\neg b2)$ は互いに疎であり，学習節 C は $C|_\pi = \neg a1 \vee a_2$ になり，$A|_\pi$ の変数のみを含む．$Pr(A|_\pi \wedge B|_\pi) = 5/8 < 7/8 = Pr(A|_\pi)$ である．

【$\because$ $A|_\pi$ は 3 変数故，$2^3 = 8$ から変数が全て偽のケースを除外すると，$Pr(A|_\pi) = (8-1)/8$．$C|_\pi$ を偽にする割り当ては $\{(a_1,1),(a_2,0),(a_3,u)\}$ 故，さらに 2 ケースを除外して，$Pr(A|_\pi \wedge B|_\pi) = (7-2)/8$.】

したがって，学習節 C が発見されると，$A|_\pi$ の異なる充足割り当ての数は 7 ではなく 5 となろう．この例で，$B|_\pi$ が，よって $F|_\pi$ が充足不能であることに注意．$F|_\pi$ が充足可能ならば，上記のような問題は生じない．

Cachet ではこの問題にいわゆる兄弟枝刈を用いて対処している．すなわち，親の成分 ψ を兄弟 $\psi|_l, \psi|_{\bar{l}}$ に分岐した結果，その一方が充足不能ならば，関係した成分は信頼できないのでキャッシュから除去する．

sharpSAT [Thurley 06] は，成分記憶を小さくする工夫等の改良により，Cachet より性能を向上させた．また，超融合 (hyper resolution) と等式簡約を導入してさらに効率を改善した研究もある [Davies 07]．

§4　阻止節のモデル計数への利用

阻止節は元来，モデル列挙のための技法であるが，モデル計数の目的にも使用できる [Morgado 05]．ただし，異なる充足部分割り当ての被覆が重ならないようにする必要がある．このため，$S_T = S \cup S_C \cup S_B$ を，元の節集合 S，矛盾節集合 S_C，阻止節集合 S_B からなる節集合とするとき，$S_T - S_C = S \cup S_B$ の節を充足するように阻止節を構成する．こうすれば，得られる新充足部分割り当ては互いに疎なモデル集合を表し，重複計数が回避できる．

例えば，入力節集合 S に対して，充足部分割り当て $\pi_1 = \{(a,1),(b,u),(c,1),(d,u)\}$，$\pi_2 = \{\ (a,u), (b,0), (c,1), (d,u)\ \}$，$\pi_3 = \{\ (a,0), (b,u), (c,u), (d,1)\ \}$，が順次得られるとしよう．$\pi_1, \pi_2$ は共に $\{\ (a,1), (b,0), (c,1), (d,0)\ \}$ を被覆するので，これらは交叉している．π_1 から阻止節 $\omega_1 = \neg a \vee \neg c$ が作られたとする．π_2 は ω_1 を充足しないので，このままでは充足部分割り当てにはなり得ない．許容される充足部分割り当ては，π_1 と交叉しない $\pi'_2 = \{(a,0),(b,0),(c,1),(d,u)\}$ である．これより，阻止節 $\omega_2 = a \vee b \vee \neg c$ が作られる．π_3 は ω_2 を充足しないので，同様に，許容される充足部分割り当てを求めると，$\pi'_{31} = \{(a,0),(b,1),(c,u),(d,1)\}$，$\pi'_{32} = \{(a,0),(b,0),(c,0),(d,1)\}$ が得られる．これらから，2 本の阻止節 $\omega_3 = a \vee \neg b \vee \neg d$，$\omega_4 = a \vee b \vee c \vee \neg d$ が作られる．

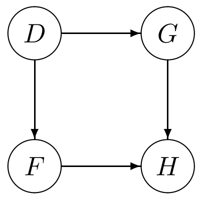
探索空間削減のため，阻止節を簡約化することを考えよう．阻止節として $S \cup S_B$ の充足に必要な決定変数割り当てだけを考慮すればよい．[Morgado 05] では，決定変数が満たすべき条件を制約方程式で表現し，このバイネイト被覆問題を解くことにより，決定変数の個数を最小化する方法が論じられている．

§5　問題表現の変換による方式

SAT では CNF が標準の問題表現形式であるが，これを別形式に変換する#SAT の手法がある．例えば，BDD に変換すると，1 の葉から根まで辿ってモデルを計数できる．一旦変換にコストをかければ，その後はデータ構造を線形時間で辿れ，記憶もわずかで済む．

$p(D)$
0.5

D	$p(G)$
真	0.7
偽	0.2



D	$p(F)$
真	0.6
偽	0.1

F	G	$p(H)$
真	真	1
真	偽	0.5
偽	真	0.4
偽	偽	0

図 5　ベイズネットの例

c2d [Darwiche 01] は CNF を DNNF(Decomposable Negation Normal Form) に変換する．DNNF は BDD を真に包含するが，より易しくなることが多い．

一般に，BDD が CNF とはかなり違う構造なのに対し，NNF は CNF に大変よく似ている．NNF は，葉が変数かその否定，葉以外は AND か OR で，深さ制限無しの根つき非循環グラフである．つまり，CNF と異なり，AND ノードと OR ノードの交代回数が任意である．

#SAT の目的に，NNF に次の二つが要求される．

分解可能性：AND ノード A が表す式を F，その子たち $A_1, \ldots, A_s$ が表す式をそれぞれ $F_1, \ldots, F_s$ とするとき，異なる F_i 間に共有変数がないこと．すなわち，子たちは完全に独立な成分であること．モデル計数は，$\#F = \#F_1 \times \ldots \times \#F_s$ で与えられる．

決定性：OR ノード O が表す式を G，その子たち $O_1, \ldots, O_t$ が表す式をそれぞれ $G_1, \ldots, G_t$ とするとき，異なる G_j 間に共有変数がないこと．すなわち，各 G_j で連言を作ると矛盾すること．モデル計数は，$\#G = \#G_1 + \ldots + \#G_t$ で与えられる．

c2d は，与えられた CNF に対し，まず全解探索 DPLL 方式により dtree と呼ばれる 2 分木を作る．各葉は CNF 中の節でタグづけされている．また，各ノードは左右部分木の共有変数の集合を保持する．その集合中の全ての変数に値を割り当てすれば左右部分木が独立となるため，分割子と呼ばれる．次に，dtree 中の分割子を順次処理しながら，DNNF を構築していく．例えば，分割子中の変数の割り当てによって成分が独立となれば，dtree 中の当該ノードに対応して AND ノードが作られる．

c2d が得意な種類の問題があり，Cachet や Relsat に優ることもある．

§6　ベイズネット推論と重み付き#SAT

ベイズネット推論も#SAT 同様，#P 問題であることが知られている．実際にベイズネットを SAT エンコーディングして解く方法も提案されている [Bacchus 03]．

図 5 にベイズネットの例を示す．図中，各ノードは真偽値をとる状態変数 D, F, G, H でラベル付けされている．各ノードに条件付確率表 (Conditional Probability Table,

状態変数（ノード名）：G, F, H
chance 変数（括弧内は重み）：$d(0.5), g_1(0.7), g_0(0.2), f_1(0.6), f_0(0.1), h_{10}(0.5), h_{01}(0.4)$
CPT に対応する節集合：
$\{ \neg d \vee \neg g_1 \vee G,\ \neg d \vee g_1 \vee \neg G,\ d \vee \neg g_0 \vee G,$
$d \vee g_0 \vee \neg G,\ \neg d \vee \neg f_1 \vee F,\ \neg d \vee f_1 \vee \neg F,$
$d \vee \neg f_0 \vee F,\ d \vee f_0 \vee \neg F,\ \neg F \vee \neg G \vee H,$
$\neg F \vee G \vee \neg h_{10} \vee H,\ \neg F \vee G \vee h_{10} \vee \neg H,$
$F \vee \neg G \vee \neg h_{01} \vee H,\ F \vee \neg G \vee h_{01} \vee \neg H,$
$F \vee G \vee \neg H \}$

図 6　ベイズネットの SAT エンコーディング例

関数 $\mathrm{BWMC}(F)$
1. もし F が空ならば，1 を返す.
2. もし F が空節を含めば，0 を返す.
3. F 中の変数 v を選ぶ;
   $\mathrm{BWMC}(F_{v=0}) \times\ weight(-v)$
   $+\mathrm{BWMC}(F_{v=1}) \times\ weight(v)$ を返す.

図 7　基本的な重み付き DPLL 型#SAT (BWMC)

CPT) が付随しており，その各行は，親ノードの状態変数の真偽値パターン（左列）に対応して，当該ノードの状態変数が真値をとる確率（右列）を与えている.

図 6 に本例の SAT エンコーディングを示す．状態変数の他に，CPT の各エントリが選択されることを表す命題変数（chance 変数と呼ぶ）$d, g_1, g_0, f_1, f_0, h_{10}, h_{01}$ が導入される．ここで，ソースノードにおける状態変数 D は chance 変数 d と同一視できるので，省かれる．これらの変数には，対応する確率が “重み” として付随する.

CPT の各行ごとに，2 本の含意式：

$$\Gamma \wedge (chance\ \text{変数}) \rightarrow X$$
$$\Gamma \wedge (chance\ \text{変数の否定}) \rightarrow \neg X$$

に対応する節を作る．ここで，X は当該ノードの状態変数，Γ は親ノードの状態変数リテラルの連言で，CPT の当該行の真偽値パターンに対応する．ただし，確率 1 の行には対応する chance 変数を設けず，$\Gamma \rightarrow X$ のみ，確率 0 の行には $\Gamma \rightarrow \neg X$ のみを作る.

本エンコーディングを，重み付 DPLL 型#SAT ソルバーで解くことにより，ベイズネット推論と等価な計算が可能となることが示されている [Sang 05].

基本的な重み付き DPLL 型#SAT (BWMC) を図 7 に示す．ただし，v が chance 変数ならば，$weight(-v) + weight(-v) = 1$，$v$ が状態変数ならば，$weight(-v) = weight(-v) = 1$ を満たす．CNF 式 F に対するある（部分）割り当てに対する重みは，その割り当てに含まれるリテラルの重みの積で得られる．s が F を充足する完全割り当てのとき，$s \models F$ と書けば，式 F の重み $weight(F)$ は，$\Sigma_{s \models F} weight(s)$ で与えられ，$weight(F) = \mathrm{BWMC}(F)$ が成り立つ.

ベイズネット N に制約 C を付加したとき，その尤度は，$\mathrm{BWMC}(F \wedge C)$ で得られる．したがって，N に関して事象 E のもとに事象 Q が生じる確率 $P(Q|E)$ が，$\frac{\mathrm{BWMC}(\varphi \wedge Q \wedge E)}{\mathrm{BWMC}(\varphi \wedge E)}$ によって求められる.

4·2　近似的なモデル計数

Walksat のような不完全型 SAT ソルバーがあるのと並行して，#SAT ソルバーにも確率統計的手法に基づくものがある.

§1　サンプリング

ApproxCount [Wei 05] は，マルコフ連鎖モンテカルロ法サンプリングに基づき，局所探索を行ってモデル数の近似値を求める．問題によっては極めて正確な近似値を与えるし，厳密なモデル計数器に比べると問題規模の増大によく耐える.

解集合からいかに均等にサンプリングできるかが鍵となる．命題式 φ が M 個の充足解をもち，これを一様ランダムにサンプリングできるとしよう．M_S 個のサンプル解の内，x が真値割り当てとなっている解の数を M_S^+ とする．比 $\gamma_S = M_S^+/M_S$ は，サンプル数の増大とともに正しい比 $\gamma = M^+/M$ に収束するだろう．ゆえに，#SAT 解を $M = (1/\gamma_S)M_S^+$ と見積ってよかろう．$1/\gamma_S$ を**倍率**と呼ぶ．こうして，$\#\varphi$ 計数問題がもっと簡単な $\varphi^+ = \varphi|_x$ のモデル計数問題に還元できる．偽値割り当ての解数が多ければ，$\varphi^- = \varphi|_{\overline{x}}$ を採ると良い．さらに，この還元を再帰的に行える．全ての変数に値が割り当てられるまで，もっと現実的には，残余式が十分小さくなって厳密計数器が適用できる程度になるまで繰り返すのである.

この解サンプリングのために，局所探索型 SAT ソルバーの Walksat を拡張して SampleSat[Wei 04] が作られた．変数選択ごとに 20-100 サンプルをとり，100-300 変数を残した段階で残余式を Relsat や Cachet などの厳密計数器にかけるのが良いようである.

サンプリング一様性の保証は難しい問題で，SampleSat では未解決である．実際，ApproxCount の計数結果は驚くほど正解に近いこともある一方，過小／過大見積の例も少なくない．最先端 SAT ソルバーは，広大な探索領域中の解を “目ざとく” 見つけられるという能力が強みなのだが，それは逆に，極めて非一様なサンプリングを行うということでもある.

Gogate らの SampleMinisat[Gogate 07] は，サンプリングの一様性を要請することなく，モデル数の下界を確率的信頼性保障付きで得る．この保障は，マルコフ不等式に基づいている．ただし，サンプリングが良くないと，得られる下界自体の品質は劣化する.

§2　XOR 制約を付加する方法

MBound [Gomes 07b] は一風変わっている．モデル数の見積に完全型 SAT ソルバーをそのまま利用する．奇数

個の変数が真ということを表す XOR 制約を，まったくランダムに生成しては次々に与式に付加し，なおも充足可能かどうかを調べる．

s 個の XOR 制約を追加してもまだ与式が充足可能なら，与式には少なくとも 2^s 個くらいのモデルがあったにちがいない．もっと正確には，s 個のランダム XOR 制約を追加する実験を t 回行っても毎回式が充足可能なら，任意の α に対し，少なくとも確率 $1-2^{-\alpha t}$ で与式が少なくとも $2^{s-\alpha}$ 個の充足割り当てを持つはずだ．実験回数 t を増やせば，あるいは，α を大きくとって弱めの下界で満足できるならば，確率の方はいくらでも高められる．

本方式の驚くべき特徴は，解の分布にまったく依存しない点である．付加した XOR 制約さえ良ければ，解集合をランダムに二つの同じくらいの集合に分割できる．実は以前，UniqueSAT（解が高々1 個であることが保障されている）問題のために提案された方式である．

§ 3　確率統計型#SAT の最新手法

BPCount [Kroc 08a] は，**確率伝播** (BP) という（確率推論の分野で “メッセージ伝播” と呼ばれる）良く知られた手続きを利用する．SAT においては，BP は変数と節との間でのある種の情報交換を意味し，相互再帰的な等式集合の形に記述して，不動点計算を行う．この不動点から式の解空間に関する統計的情報が容易に求まる．与式から導かれる制約グラフが木構造ならば，得られる統計的情報は正確で，循環があっても誤差はたいてい小さい．

BP は，原理的に SampleCount[Gomes 07a] の解サンプルから得られるものとちょうど同じ情報を与える．すなわち，解が一様ランダムにサンプリングされたときの各変数の真偽値の確率である．しかるに，BPCount は SampleSat より格段に速い．問題は，実践的な命題式に対し，BP の相互再帰式が不動点に収束しない場合が結構多いことだ．この改善のため，Kroc らはメッセージダンピングなる BP の変形を考案している．

SampleCount の下界解析ではマルコフ不等式が有効だが，上界に対してはそうはいかない．Kroc らの MiniCount[Kroc 08b] は，MiniSat に 2 つの修正を施して解数を見積もる方法を与えた．変数の真偽値割り当てのランダム化，および再計算の禁止である．解に至るまでの分岐回数 d に注目する．d の期待値 $\mathbf{E}[d]$ を効率的に見積りたい．割り当てがランダムなら，$\mathbf{E}[d]$ はモデル数の $\log_2$ より小さくはないだろう．MiniSat をランダム割り当てモードで複数回実行し，d の平均をとってみる．難しい解（d 大）に当たる頻度は，易しい解（d 小）より小さいだろう．すると，$\mathbf{E}[d]$ が小さすぎ，間違った上界を与えてしまうかもしれない．

ところが実は，多くの場合，d は正規分布に非常に近い．つまり，上界の期待値 2^d は対数正規分布に近い．それならば，必ずしも d の低めの値や高めの値を必要としない．d の平均以下のサンプルのみですら十分である．正規性の仮定により，標準的な統計的計算を用いて任意の信頼区間内で $\mathbf{E}[d]$ の上界が得られる．

こうして，MiniCount は多くの問題に対し，数秒以内の早さで非常に良い上界（正解，もしくは下界の値に極めて近い）を与える．

5. お わ り に

本稿では，SAT 問題の中でも，全解探索が基本のモデル列挙やモデル計数に焦点を当てた技術項目について解説した．与式のモデルの個数を計数する#SAT は，式の充足可能性を判定する SAT よりも難しい問題であり，技術的な課題も多く，今後さらなる研究の余地が残されている．モデル列挙にしろモデル計数にしろ節学習が性能改善の鍵を握るが，モデル計数に阻止節学習を組込む効率の良い方式は確立されていない．この代替として，極小モデルや極小限定をモデル計数に利用する方式も考えられる．

本稿で触れなかった項目については，[Gomes 09] に包括的な記述があるので，そちらを参照されたい．

◇ 参 考 文 献 ◇

[Bacchus 03] Bacchus, F., Dalmao, S., , and Pitassi, T.: Algorithms and Complexity Results for #SAT and Bayesian Inference, in *Proceedings of FOCS-03: 44th Annual Symposium on Foundations of Computer Science*, pp. 340–351 (2003)

[Bayardo 00] Bayardo, R. J. and Pehoushek, J. D.: Counting Models using Connected Components, in *Proceedings AAAI-00: 17th National Conference on Artificial Intelligence*, pp. 157–162 (2000)

[Biere 99] Biere, A., Cimatti, A., Clarke, E., and Zhu, Y.: Symbolic Model Checking without BDDs, in *Proceedings of TACS'99: Fifth International Conference on Tools and Algorithms for Construction and Analysis of Systems*, pp. 193–207 (1999)

[Birnbaum 99] Birnbaum, E. and Lozinskii, E. L.: The Good Old Davis-Putnam Procedure helps Counting Models, *Journal of Artificial Intelligence Research*, Vol. 10, pp. 457–477 (1999)

[Bry 96] Bry, F. and Yahya, A.: Minimal Model Generation with Positive Unit Hyper-Resolution Tableaux, in Miglioli, et al. [Miglioli 96], pp. 143–159

[Darwiche 01] Darwiche, A.: Decomposable Negation Normal Form, *Journal of the ACM*, Vol. 48, No. 4, pp. 608–647 (2001)

[Darwiche 04] Darwiche, A.: New Advances in Compiling CNF into Decomposable Negation Normal Form, in *Proceedings of ECAI-04: 16th European Conference on Artificial Intelligence*, pp. 328–332 (2004)

[Davies 07] Davies, J. and Bacchus, F.: Using More Reasoning to Improve #SAT Solving, in *Proceedings of AAAI-07: 22nd National Conference on Artificial Intelligence*, pp. 185–190 (2007)

[Davis 60] Davis, M. and Putnam, H.: A Computing Procedure for Quantification Theory, *Journal of ACM*, Vol. 7, pp. 201–215 (1960)

[Davis 62] Davis, M., Logemann, G., and Loveland, D.: A Machine Program for Theorem Proving, *Communications of ACM*, Vol. 5, pp. 394–397 (1962)

[Eén 05] Eén, N. and Sörensson, N.: MiniSat: A SAT Solver with Conflict-Clause Minimization, in *Proceedings of SAT-05: 8th International Conference on Theory and Applications of Satisfiability Testing*, pp. 502–518 (2005)

[Gogate 07] Gogate, V. and Dechter, R.: Approximate counting by sampling the backtrackfree search space, in *Proc. of AAAI-07: 22nd National Conference on Artificial Intelligence*, pp. 198–203 (2007)

[Gomes 07a] Gomes, C. P., Hoffmann, J., Sabharwal, A., , and Selman, B.: From sampling to model counting, in *Proc. of IJCAI-07:*

20th International Joint Conference on Artificial Intelligence, pp. 2293–2299 (2007)

[Gomes 07b] Gomes, C. P., Hoffmann, J., Sabharwal, A., and Selman, B.: Short XORs for Model Counting: From Theory to Practice, in *Proceedings of SAT-07: 10th International Conference on Theory and Applications of Satisfiability Testing*, pp. 100–106 (2007)

[Gomes 09] Gomes, C. P., Sabharwal, A., and Selman, B.: Model Counting, in *Handbook of Satisfiability*, chapter 20, IOS Press (2009)

[Grove 94] Grove, A. J., Halpern, J. Y., and Koller, D.: Random Worlds and Maximum Entropy, *Journal of Artifitial Intelligence Research*, Vol. 2, pp. 33–88 (1994)

[Hasegawa 00] Hasegawa, R., Fujita, H., and Koshimura, M.: Efficient Minimal Model Generation Using Branching Lemmas, in *Proceedings of 17th International Conference on Automated Deduction, CADE-17*, LNAI 1831, pp. 184–199, Springer (2000)

[長谷川 01] 長谷川 隆三, 藤田 博, 越村 三幸 : 分岐補題の抽出による極小モデル生成の効率化, 人工知能学会論文誌, Vol. 16, No. 2, pp. 234–245 (2001)

[長谷川 08] 長谷川 隆三, 藤田 博, 越村 三幸：モデル生成型定理証明と要素技術, コンピュータソフトウェア, Vol. 25, No. 3, pp. 2–10 (2008)

[井上 08] 井上 克巳, 坂間 千秋：論理プログラミングから解集合プログラミングへ, コンピュータソフトウェア, Vol. 25, No. 3, pp. 20–32 (2008)

[Jin 05] Jin, H., Han, H., and Somenzi, F.: Efficient Conflict Analysis for Finding All Satisfying Assignments of a Boolean Circuit, in *Proceedings of TACAS 2005: 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 287–300 (2005)

[Kauz 96] Kauz, H. and Selman, B.: Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search, in *Proceedings of AAAI-96: 13th National Conference on Artificial Intelligence*, pp. 1194–1201 (1996)

[Koshimura 03] Koshimura, M., Iwaki, M., and Hasegawa, R.: Minimal Model Generation with Factorization and Constrained Search, 情報処理学会論文誌, Vol. 44, No. 4, pp. 1163–1172 (2003)

[越村 08] 越村 三幸, 鍋島 英知, 藤田 博, 長谷川 隆三：極小モデル生成とジョブショップスケジューリング問題の解法, 日本ソフトウェア科学会第 25 回大会 講演論文集 (2008), 8A-2

[Kroc 08a] Kroc, L., Sabharwal, A., and Selman, B.: Leveraging Belief Propagation, Backtrack Search, and Statistics for Model Counting, in *Proceedings of CPAIOR-08: 5th International Conference on Integration of AI and OR Techniques in Constraint Programming*, pp. 127–141 (2008)

[Kroc 08b] Kroc, L., Sabharwal, A., and Selman, B.: Leveraging belief propagation, backtrack search, and statistics for model counting, in *Proc. of CPAIOR-08: 5th International Conference on Integration of AI and OR Techniques in Constraint Programming*, pp. 127–141 (2008)

[Letz 94] Letz, R., Mayr, K., and Goller, C.: Controlled Integration of the Cut Rule into Connection Tableau Calculi, *Journal of Automated Reasoning*, Vol. 13, No. 3, pp. 297–337 (1994)

[McMillan 02] McMillan, K. L.: Applying SAT Methods in Unbounded Symbolic Model Checking, in *Proceedings of CAV 2002: Computer Aided Verification*, pp. 250–264 (2002)

[Miglioli 96] Miglioli, P., Moscato, U., Mundici, D., and Ornachi, M. eds.: *Proceedings of the 5th International Workshop on Theorem Proving with Analytic Tableaux and Related Methods, TABLEAUX'96*, LNAI 1071,Springer (1996)

[Morgado 05] Morgado, A. and Marques-Silva, J.: Good Learning and Implicit Model Enumeration, in *Proceedings of International Conference on Tools with Artificial Intelligence*, pp. 131–136 (2005)

[Moskewicz 01] Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S.: Chaff: Engineering an Efficient SAT Solver, in *Proceedings of Design Automation Conference*, pp. 530–535 (2001)

[Niemelä 96] Niemelä, I.: A Tableau Calculus for Minimal Model Reasoning, in Miglioli, et al. [Miglioli 96], pp. 278–294

[Roth 96] Roth, D.: On the Hardness of Approximate Reasoning, *Artificial Intelligence*, Vol. 82, pp. 273–302 (1996)

[Sang 04] Sang, T., Bacchus, F., Beame, P., Kautz, H., and Pitassi, T.: Combining Component Caching and Clause Learning for Effective Model Counting, in *Proceedings of SAT-04: 7th International Conference on Theory and Applications of Satisfiability Testing*, pp. 20–28 (2004)

[Sang 05] Sang, T., Beame, P., and Kautz, H.: Solving Bayesian Networks by Weighted Model Counting, in *Proceedings of AAAI-05: 20th National Conference on Artificial Intelligence*, pp. 475–482 (2005)

[Selman 93] Selman, B., Kautz, H., and Cohen, B.: Local Search Strategies for Satisfiability Testing, in *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, Vol. 26 of *Series in Discrete Mathematics and Theoretical Computer Science*, pp. 521–531 (1993)

[Thurley 06] Thurley, M.: sharpSAT - Counting Models with Advanced Component Caching and Implicit BCP, in *Proceedings of SAT-06: 9th International Conference on Theory and Applications of Satisfiability Testing*, pp. 424–429 (2006)

[Valiant 79] Valiant, L. G.: The Complexity of Computing the Permanent, *Theoretical Computer Science*, Vol. 8, pp. 189–201 (1979)

[Wei 04] Wei, W., Erenrich, J., and B.fSelman, : Towards effcient sampling: Exploiting random walk strategies, in *Proc. of AAAI-04: 19th National Conference on Artificial Intelligence*, pp. 670–676 (2004)

[Wei 05] Wei, W. and Selman, B.: A New Approach to Model Counting, in *Proceedings of SAT-05: 8th International Conference on Theory and Applications of Satisfiability Testing*, pp. 324–339 (2005)

〔担当委員：××○○〕

19YY 年 MM 月 DD 日　受理

著　者　紹　介

長谷川　隆三（正会員）

1972 年九州大学工学部通信工学科卒．1974 年同大学大学院工学研究科通信工学専攻修士課程修了．同年日本電信電話公社（現 NTT）入社．1987 年（財）新世代コンピュータ技術開発機構に出向．1995 年九州大学工学部教授．現在，同大学大学院システム情報科学研究院教授．九州大学博士（工学）．人工知能，自動推論，高並列計算機方式に関する教育研究に従事．

藤田　　博（正会員）

1978 年東京大学理学部物理学科卒．1980 年同大学大学院工学系研究科情報工学専攻修士課程修了．同年三菱電機（株）入社．1986〜1990 年（財）新世代コンピュータ技術開発機構に出向．1996 年九州大学大学院システム情報科学科助教授．現在，同大学大学院システム情報科学研究院准教授．東京大学博士（工学）．プログラミング，自動推論に関する教育研究に従事．

越村　三幸（正会員）

1984 年筑波大学第一学群自然学類卒．1986 年同大学大学院修士課程理工学研究科修了．同年日本ビジネスオートメーション（株）（現 東芝情報システム（株））入社．1986〜1990，1993〜1995 年（財）新世代コンピュータ技術開発機構に出向．1995 年九州大学工学部助手．現在，同大学大学院システム情報科学研究院助教．九州大学博士（工学）．自動推論システムに関する教育研究に従事．