

モデル生成による SAT ソルバの並列化

矢野 明浩† 越村 三幸‡ 藤田 博‡ 長谷川 隆三‡

†九州大学 電気情報工学科

‡九州大学 システム情報科学研究院

1 はじめに

充足可能性問題 (SATisfiability problem, SAT) とは、与えられた論理式に対し、式全体を真とする変数の割り当てが存在するか否かを判定する問題である。ソフトウェア・ハードウェア検証や人工知能設計などは SAT 問題への多項式時間還元が可能であることが知られており、それを解く方が変換前の問題を解くより簡単であることが多い。このため、SAT 問題をより高速に解くことができるプログラム (SAT ソルバ) を作成することには意義があると言える。

既存の SAT ソルバの 1 つに MiniSat[1] と呼ばれるものがある。これは、ヒューリスティックによる変数選択を用いた探索・矛盾学習機能・リスタートを備えており、様々な大会で優秀な成績を納めている。

本論文では、まず第 2 節で SAT 問題と MiniSat の簡単な説明を行う。次に、第 3 節ではこの MiniSat に手を加え更なる高速化を行うための手法を提案する。また、与えられた問題中に存在する並列性に着目し、従来と異なる並列化の実装について検討する。

2 SAT 問題

2.1 SAT 問題とは

SAT 問題とは与えられた命題論理式が真となる (SAT) 変数の割り当てを発見するか、どのような変数割り当てを行っても真とすることができない (UNSAT) ことを示す問題である。問題はリテラル (literal) の論理和からなる節 (clause) の論理積で与えられる。この形式を CNF (Conjunctive Normal Form) と呼ぶ。CNF の例を下に示す。

$$F_1 = (a \vee b \vee c) \wedge (\neg a \vee b) \wedge (\neg b \vee \neg c)$$

$$F_2 = \neg a \wedge (a \vee b) \wedge (a \vee \neg b)$$

F_1 は $\{a = \text{True}, b = \text{True}, c = \text{False}\}$ と割り当てることにより全体を真とすることができるため、SAT である。

F_2 は変数 a, b にどのような割り当てを行っても全体を真とすることができないため、UNSAT である。

2.2 MiniSat

MiniSat に含まれるいくつかの機能のうち第 3 節で取り上げる以下のものに関して簡単な説明を行う。

- 変数決定
- バックトラック

MiniSat における変数決定は、VSIDS (Variable State Independent Decaying Sum) に基づいて行われる。これは、各変数に対し activity という値を与え、この値が大きいのを優先的に選択する。activity は、変数割り当てに矛盾が発生した際にそれに関与した変数について増加させる。また、この値は一定時間ごとに減少させる。これにより、最近よく使われる変数の activity が大きくなり、実行時間の短縮につながる良い変数選択を行うことができる。

従来の SAT ソルバでは、変数割り当て中に矛盾が発生した場合、割り当てを一段階前の状態に戻し、探索を再開する。一方 MiniSat では、矛盾が発生するとその原因を解析し、数段階前の状態まで戻す。このため、ソルバが意味のない領域を探索する前に、そのような領域を刈り取ることができる。このようなバックトラックを非時間順バックトラック (Non-chronological Backtracking) と呼ぶ。

2.3 極小モデル生成

MiniSat の実行は単解探索である。本論文では、これを全解探索とし、その際に生成するモデルについては他のモデルに包摂されないもののみを対象とする。このようなモデルを極小モデルと呼ぶ。極小モデルを発見するために、負リテラル優先分岐と以下に述べるモデル制約を導入する [2]。

3 Mini-MG

今回発表者の研究室では、MiniSat を元にモデル生成法に基づく新たな SAT ソルバを開発し Mini-MG と名づけた。

モデル生成法とは、与えられた節集合を充足する変数割り当てを構成する方法である。空モデル M から始

め、M で充足されない違反節を充足するように M を拡張していく。この実現に必要な技術を本節で述べる。

3.1 Complement Splitting

与えられた CNF に含まれる各節を以下の 3 つに区別する。

正節: 全てのリテラルが正である節

負節: 全てのリテラルが負である節

混合節: 上記以外の節

また、混合節のうち含まれる負リテラルが全て偽に割り当てられているものを違反節と呼ぶ。以下に例を挙げる。

$$MCI = a \vee b \vee \neg c \vee \neg d$$

割り当て: $c = \text{True}, d = \text{True}$

本論文では、正節と違反節に含まれる全ての正リテラルを元に探索の分岐を行う。

ある正節 (違反節) を選び、そこに含まれている正リテラルを取り出したものが

$$C = B_1 \vee B_2 \vee, \dots, \vee B_n$$

で与えられたとする。探索の分岐は図 1 のように行う。

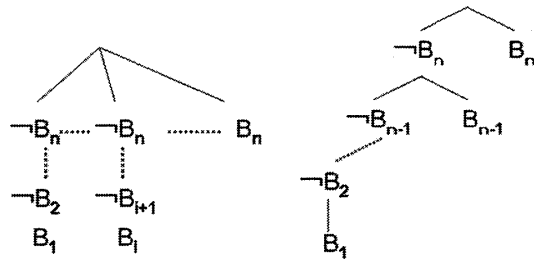


図 1: complement splitting 図 2: complement splitting2

図 1 について、各枝は探索領域を重複なく分割している。このような分割を相補分割 (complement splitting) と呼ぶ。今回の Mini-MG では MiniSat に習って探索を行なうために、図 2 のような分岐枝を作り実行することとした。分割された領域は互いに重複しないため、並列化の際にも利用しやすい。

3.2 バックトラック

SAT ソルバ実行中にバックトラックが必要となることがあるが、Mini-MG では矛盾が発生した場合と極小モデルを発見した場合に分けて考える。

探索中に矛盾が発生した場合は、通常の MiniSat のバックトラックをそのまま用いる。これにより、矛盾解析と学習による MiniSat の高速性をそのまま生かすことができる。

探索により極小モデルを発見した場合、次に探索するのは complement splitting による次の枝である。このため、バックトラックをする際には全ての分岐枝とこれから分岐を行う正節・違反節の情報を各探索レベル (深さ) 別に蓄えておく必要がある。

モデルを発見した際は、そのレベルから順にスタックに保持された決定変数を参照し、それが負リテラルになるまで 1 レベルずつバックトラックを行う。決定変数が負リテラルであるレベルまでバックトラックが終了すると、その決定変数を正リテラルにフリップし再度探索を行う。

3.3 モデル制約

全解探索を行なうためには、一度発見したモデルを二度発見しないように工夫を行なう必要がある。

今回は発見した極小モデルに含まれる正リテラル全てからなる部分集合の各正リテラルの補をとり新たな節として問題に追加することで発見済みにモデルを再度発見することを回避するようにした。

4 おわりに

今回の論文では、MiniSat を高速性を生かした新たな SAT ソルバを作成する際に必要となる技術を並列化を視野にいれて提案した。従来の並列 SAT ソルバは、変数に注目した領域分割を行うものがほとんどである。それに対し今回提案した complement splitting は問題に出現する節に注目した新しい領域分割である。今後さらに Mini-MG の有効性を検証し、並列化の実装も行なっていきたい。

謝辞 本研究は科研費 20240003 の助成を受けたものである。

参考文献

- [1] Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502-518. Springer, Heidelberg (2004)
- [2] 長谷川隆三, 藤田博, 越村三幸: モデル列挙とモデル計数, 人工知能学会誌, Vol.25, No.1, pp.96-104 (2010)