1E3-OS-4

効率的なSMTソルバの実現を目指した等号組み合わせの削減

Reduction of equality constraints for realizing an efficient SMT solver

福田　寿志[*1]　岩沼　宏治[*2]　山本　泰生[*2]
Hisashi Fukuda　Koji Iwanuma　Yoshitaka Yamamoto

[*1]山梨大学大学院医学工学総合教育部コンピュータ・メディア工学専攻
Department of Computer Science and Media Engineering, Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

[*2]山梨大学大学院コンピュータ・メディア工学専攻
Department of Computer Science and Media Engineering, University of Yamanashi

SMT is an extended SAT technology in order to process the background theory described in the first-order logic. Practical background theories usually consist of two or more first-order theories.The Nelson-Oppen method creates constraints, in order to prevent the inconsistency occurring between determination procedure is. In this paper, we show an implementation of a SMT solver which has mechanism for reducing equality constraint formulas which are automatically generated for preventing inconsistency decisions.

1. はじめに

近年の命題論理の**充足可能性判定問題** (SAT 問題) を解く技術の進展に伴い，SAT を用いたソフトウェアやハードウェアの形式的検証やプランニング問題，スケジューリング問題などさまざまな分野において盛んに研究されている．一般にこれらの検証・推論には，等号や算術などに関する背景知識が必要となる．実用問題では，複雑な背景知識が必要になるめ，命題論理より表現能力の高い一階論理などの論理体系で背景知識を記述できれば，記述がコンパクトで簡潔になり，都合がよいことが多い．

背景理論付き SAT(Satisfiability Modulo Theories: SMT)[3][4] とは，このような命題論理よりも表現能力の高い論理体系で記述された背景理論を SAT 技法で効果的に取り扱うことを目的とした技術である．SMT の実用問題で必要となる大規模で複雑な背景知識は複数の理論から構成されいるが，**Nelson-Oppen 法** [1] はそれら複数理論の個々の決定手続きを組み合わせて判定を可能にするための理論である．Nelson-Oppen 法では，組み合わせる決定手続き間での矛盾の発生を防ぐために等号の制約を問題式に追加する．本研究では，効率的な SMT ソルバの実現のため，制約条件の削減機能を持つ SMT ソルバの実装を行ったので，その報告を行う．

2. SAT 問題

SAT 問題を定義する．SAT 問題は通常**連言標準形 (conjunctive normal form: CNF)** で与えられる．CNF は，命題または命題の否定を表すリテラル (literal) の選言の連言である．SAT 問題を解くとは，全ての節を充足する命題の真偽値割り当てを求めることである．もしそのような真偽値割り当てが 1 つ以上存在するならば充足可能 (SAT) であり，そうでない場合，充足不可能 (UNSAT) と言う．

例えば，次のような CNF 式 $(\neg A \vee \neg B) \wedge (B \vee C) \wedge A$ が与えられたとき，$A : True$, $B : False$, $C : True$ と割り当てることで式を充足できる．

連絡先: 福田寿志，山梨大学大学院医学工学総合教育部コンピュータメディア工学専攻，g11mk032@yamanashi.ac.jp

3. SMT 技術

SMT 技術は**事前処理型 SMT 技術**と**遅延処理型 SMT 技術**の 2 つに大別される [4]．事前処理型 SMT 技術とは，背景知識や質問を事前に命題論理式へコンパイル・符号化し，既存の高速 SAT ソルバで最終的に解かせるものである．常に最新の高性能 SAT ソルバを利用できるメリットがある反面，多くの背景理論に対して符号化方式を個別に作成する必要がある．また符号化出力した命題式は巨大なものになることも多く，メモリオーバなどの問題に直面することも多い．

これに対して遅延処理型 SMT 技術は，個別の理論に特化した既存ソルバと最新 SAT ソルバ技術を効果的に組み合わせる目的で発展してきた技術である．

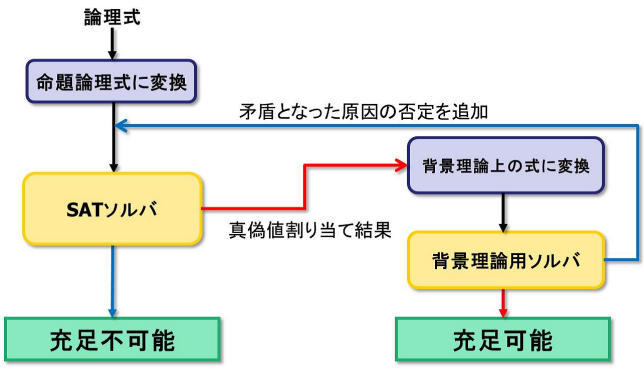

図 1: 遅延処理型 SMT ソルバの概要

遅延処理型 SMT 技術では，図 1 に示すように，充足可能性を判定する一階論理式は，まず命題式として取り扱われ，SAT ソルバで処理される．もし SAT ソルバで充足不可能と判定されれば，処理はそこで終了する．充足可能と判定された場合，SAT ソルバから命題論理モデルが出力されるので，一階論理に復元され，背景理論の専用ソルバが起動し，背景理論との無矛盾性を調べる．無矛盾であると判定されれば，そこで処理が

終了する．矛盾していれば，SAT ソルバが別の命題論理モデルの探索を再開し，再度同じことを繰り返す．柔軟性に富む枠組みであり，全体として高速といわれている．

本論文では，遅延処理型 SMT 技術に関して議論していく．

4. SMT 背景理論の決定手続きの結合法

SMT の実用問題で必要となる大規模で複雑な背景理論は，複数の背景理論が組み合わされて構成されている．それぞれの背景理論に対しては，長年の研究に基づく非常に精密で高度な決定手続きとそれを実装した複雑なプログラム (専用ソルバ) が用意されている．そのため，実用問題であつかうような複合型理論に対して新たに専用ソルバを開発するのは大変な苦労が伴い，応用性も低い．そのため，個々の背景理論に対して開発された既存の専用ソルバをそのまま組み合わせて利用する手法が提案されている．ここでは，その代表例とその実装技術について紹介する．

4.1 Nelson-Oppen 法

Nelson-Oppen 法 [1] は，等号以外は互いに共通記号を持たない背景理論 T_1, T_2 に対して，その合併理論 $T_1 \wedge T_2$ を考え，基礎式 F が与えられたときに，F が $(T_1 \wedge T_2)$-充足可能か否かを新たに $(T_1 \wedge T_2)$-ソルバを作らずに，既存の T_1-ソルバと T_2-ソルバだけを使って判定を可能にするものである [4]．この手法は，30 年以上前に提案されたものであるが，これを本質的に超えるものはまだ存在していない．

次節にて Nelson-Oppen 法の実装手法である Delayed Theory Combination の紹介を行う．

4.2 Delayed Theory Combination とその実装

Nelson-Oppen 法では，背景理論用ソルバを独立に動かし判定を進めるため，各背景理論用ソルバで矛盾がおこらないように制約条件を作成する． 既存の背景理論用ソルバは前節で述べたように極めて精巧に作られており，改造は非常に難しいため，各背景理論用ソルバは改造せずにそのまま利用したい．また，制約条件を，実行中に動的に生成することは手間がかかるので，可能であれば，回避したい．そのための改良案として **Delayed Theory Combination(DTC)**[5] が開発された．

この手法のアルゴリズムを，以下の図 2 に示す．

function Bool+T_1+T_2 (ϕ: *quantifier-free formula*)
1 $\phi \longleftarrow purify(\phi)$;
2 $\phi^p \longleftarrow fol2prop(\phi)$; $\mathcal{A}^p \longleftarrow fol2prop(Atoms(\phi) \cup IE(\phi))$;
3 **while** Bool-*satisfiable* (ϕ^p) **do**
4 $\beta_1^p \wedge \beta_2^p \wedge \beta_e^p = \beta^p \longleftarrow pick\_total\_assign(\mathcal{A}^p, \phi^p)$;
5 $(\rho_1, \pi_1) \longleftarrow T_1$-*satisfiable* ($prop2fol(\beta_1^p \wedge \beta_e^p)$);
6 $(\rho_2, \pi_2) \longleftarrow T_2$-*satisfiable* ($prop2fol(\beta_2^p \wedge \beta_e^p)$);
7 **if** ($\rho_1 ==$ sat **and** $\rho_2 ==$ sat) **then return** sat;
8 **if** ($\rho_1 ==$ unsat) **then** $\phi^p \longleftarrow \phi^p \wedge \neg fol2prop(\pi_1)$;
9 **if** ($\rho_2 ==$ unsat) **then** $\phi^p \longleftarrow \phi^p \wedge \neg fol2prop(\pi_2)$;
10 **end while**;
11 **return** unsat;
end function

図 2: DTC を用いた 2 つの背景理論に対する SMT ソルバのアルゴリズム [5]

次に，このアルゴリズムの説明をかねて動作例を示す．以下の未定関数記号 f をもつ等式理論 (T_E) と算術理論 (T_Z) からなる式 ϕ の充足可能性を判定する．

$$\phi : 1 \leq a \wedge a \leq 2 \wedge f(a) \neq f(1) \wedge f(a) \neq f(2)$$

この式 ϕ は理論 $T_E \wedge T_Z$ のもとで充足不可能である．

まず，$purify(\phi)$ で式を扱いやすい形に変形する．

これは，DTC が背景理論用ソルバを独立して適用するために，式 ϕ を単一の背景理論からなる式に分ける．その際，未定関数記号 f は T_E で扱われ，定数 $1, 2$ は T_Z で扱われるため，それらが同じ項に存在する $f(1)$ と $f(2)$ は式を分割する際に都合が悪い．そこで，定数 $1, 2$ を変数 w_1, w_2 で置き換えて式を純化し，分けやすくする．

純化を行った式は以下のようになる．

$$\phi : 1 \leq a \wedge a \leq 2 \wedge f(a) \neq f(w_1) \wedge f(a) \neq f(w_2) \wedge w_1 = 1 \wedge w_2 = 2$$

次に $Atoms(\phi)$ で式 ϕ の項を取得し，$IE(\phi)$ で制約条件を作成する．その結果が以下のようになる．

$$Atoms(\phi) = \{1 \leq a, a \leq 2, f(a) = f(w_1), f(a) = f(w_2), w_1 = 1, w_2 = 2\}$$
$$IE(\phi) = \{a = w_1, a = w_2, w_1 = w_2\}$$

さらにこれらを $fol2prop$ を用いて命題化する．この結果を A^p として保持する．これは式 ϕ の命題充足可能性を判定した後に一階理論に復元する際に用いる．

$$A^p = \{A, B, C, D, E, F, G, H, I\}$$

例の後半で使用するため，命題化した変数ともとの項との関係を詳しく記すと，A は $1 \leq a$，B は $a \leq 2$，C は $f(a) = f(w_1)$，D は $f(a) = f(w_2)$，E は $w_1 = 1$，F は $w_2 = 2$，G は $a = w_1$，H は $a = w_2$，I は $w_1 = w_2$，を命題化したものである．

次に，式 ϕ の命題充足可能性を判定する．そのため，式 ϕ の各項を先ほど対応付けた命題変数で置き換える．置き換えた結果が以下のようになる．

$$\phi^p : A \wedge B \wedge \neg C \wedge \neg D \wedge E \wedge F$$

この式を SAT ソルバを用いて判定する．

充足不能と判定された場合，式 ϕ は充足不可能である．充足可能と判定された場合は以下の処理を繰り返す．

SAT ソルバから返された真偽値割り当てをもとに一階理論の式に復元する．今回，C, D に $False$ が割り当てられ，それ以外に $True$ が割り当てられたとする．このとき式 ϕ^p が充足可能と判定されるので，この割り当てから A^p をもとに一階論理式へ復元 (5, 6 行目の $prop2fol$) し，各背景理論からなる式と共通変数への制約に分割する．

$$\beta_E : f(a) \neq f(w_1) \wedge f(a) \neq f(w_2)$$
$$\beta_Z : 1 \leq a \wedge a \leq 2 \wedge w_1 = 1 \wedge w_2 = 2$$
$$\beta_e : a = w_1 \wedge a = w_2 \wedge w_1 = w_2$$

これを各背景理論用のソルバに制約条件と一緒に渡し，充足可能性を判定する．T_Eソルバ には，$\beta_E \wedge \beta_e$ を，T_Zソルバ には，$\beta_z \wedge \beta_e$ を渡して判定する．

その結果，上の条件では充足不能であると判定される．その際，充足不能の原因として T_Eソルバ から $f(a) \neq f(w_1) \wedge a = w_1$ と $f(a) \neq f(w_2) \wedge a = w_2$ が，T_Zソルバ から $w_1 = 1 \wedge w_2 = 2 \wedge w_1 = w_2$ が得られる．

次に, その否定を式に付加 (8, 9 行目の ¬ $fol2prop$) し, もう一度同じ動作を行う.

$$\phi^p : A \wedge B \wedge \neg C \wedge \neg D \wedge E \wedge F \wedge \neg(\neg C \wedge G) \wedge \neg(\neg D \wedge H) \wedge \neg(E \wedge F \wedge I)$$

今度は SAT ソルバが C, D, G, H, I に $False$, それ以外に $True$ を割り当てたとする. この割り当てをもとに再度一階論理式へ復元し, 各背景理論からなる式と制約条件に分割する.

$$\beta_E : f(a) \neq f(w_1) \wedge f(a) \neq f(w_2)$$
$$\beta_Z : 1 \leq a \wedge a \leq 2 \wedge w_1 = 1 \wedge w_2 = 2$$
$$\beta_e : a \neq w_1 \wedge a \neq w_2 \wedge w_1 \neq w_2$$

先ほどと同様に各背景理論用のソルバに渡し, 充足可能性を判定する. その結果, 今回も充足不能であると判定される.

その際, 充足不能の原因として T_Z ソルバ から $1 \leq a \wedge a \leq 2 \wedge w_1 \leq 1 \wedge w_2 \leq \wedge a \neq w_1 \wedge a \neq w_2$ が得られる. そのため, その否定を式に付加し, もう一度同じ動作を繰り返す.

$$\phi^p : A \wedge B \wedge \neg C \wedge \neg D \wedge E \wedge F \wedge \neg(\neg C \wedge G) \wedge \neg(\neg D \wedge H) \wedge \neg(E \wedge F \wedge I) \wedge \neg(A \wedge B \wedge E \wedge F \wedge \neg G \wedge \neg H)$$

今度は SAT ソルバが上の式は充足不可能であると返すので, もとの式 μ は充足不能であると判定される.

このようすることで, DTC では問題式の判定を各ソルバを完全に独立させて行うことができる. 現在, この手法の実装を行っている.

実装内容の概要を図 3 に示す.

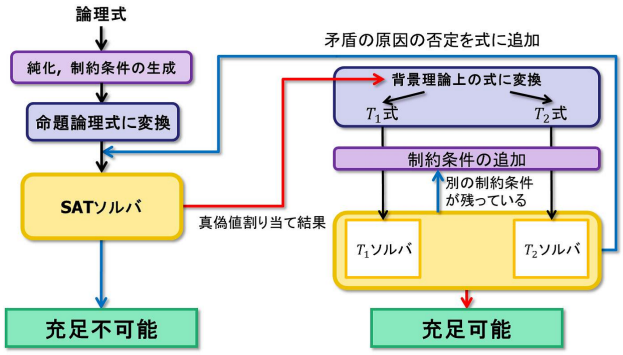

図 3: 実装内容の概要

図 3 のように上で紹介したアルゴリズムとは少し異なる. 先ほどのアルゴリズムでは, 制約条件における 2 変数間の等号, 不等号は SAT ソルバからの真偽値割り当てをもとに決めらる. そのため, 真偽値割り当て毎に一階論理式に復元する必要があり手間となる場合がある. そこで, 考えられる制約条件を全種類作成しておき, 問題式の真偽値割り当てを復元した式に 1 つずつ追加し判定を行うことで復元回数を減らす.

異なる点を先ほどと同じ式 φ の判定を行う動作をもとに説明する.

まず, 以下に示すように純化した式 φ をもとに, 共通変数とそこから考えられるすべての制約条件を作成する.

$$\phi : 1 \leq a \wedge a \leq 2 \wedge f(a) \neq f(w_1) \wedge f(a) \neq f(w_2) \wedge w_1 = 1 \wedge w_2 = 2$$

共通変数

$$a, w_1, w_2$$

制約条件

$$C_1 : a = w_1 \wedge a = w_2 \wedge w_1 = w_2$$
$$C_2 : a = w_1 \wedge a = w_2 \wedge w_1 \neq w_2$$
$$C_3 : a = w_1 \wedge a \neq w_2 \wedge w_1 = w_2$$
$$\cdot$$
$$\cdot$$
$$C_8 : a \neq w_1 \wedge a \neq w_2 \wedge w_1 \neq w_2$$

次に, 式 φ の各リテラルを命題変数で置き換えた式 ϕ^p を作成し, 式 φ の命題充足可能性を判定する.

$$\phi^p : A \wedge B \wedge C \wedge D \wedge E \wedge F$$

その真偽値判定結果をもとに元の一階論理式に復元する. 今回すべての命題変数に真が割り当てられたとすると, 復元した式 φ は以下のようになる.

$$\phi : 1 \leq a \wedge a \leq 2 \wedge f(a) \neq f(w_1) \wedge f(a) \neq f(w_2) \wedge w_1 = 1 \wedge w_2 = 2$$

この式 φ を各背景理論からなる式 (ϕ_E, ϕ_Z) に分割する.

$$\phi_E : f(a) \neq f(w_1) \wedge f(a) \neq f(w_2)$$
$$\phi_Z : 1 \leq a \wedge a \leq 2 \wedge w_1 = 1 \wedge w_2 = 2$$

分割した各式に, 先ほど作成した制約条件を 1 つずつ追加し判定を行う. まず, 制約条件 C_1 を各式に追加する.

$$\phi_{E,C_1} : f(a) \neq f(w_1) \wedge f(a) \neq f(w_2) \wedge a = w_1 \wedge a = w_2 \wedge w_1 = w_2$$

$$\phi_{Z,C_1} : 1 \leq a \wedge a \leq 2 \wedge w_1 = 1 \wedge w_2 = 2 \wedge a = w_1 \wedge a = w_2 \wedge w_1 = w_2$$

この式を専用ソルバを用いて判定する. ここで, 式 ϕ_{E,C_1} は項 $f(a) \neq f(w_1)$ と $a = w_1$ に矛盾が生じるため UNSAT となる. そこで, 先ほどの式 ϕ_E, ϕ_Z に別の制約条件を追加し判定を続けていく. ここでは, 制約条件 C_2 の追加を行う.

$$\phi_{E,C_2} : f(a) \neq f(w_1) \wedge f(a) \neq f(w_2) \wedge a = w_1 \wedge a = w_2 \wedge w_1 \neq w_2$$

$$\phi_{Z,C_2} : 1 \leq a \wedge a \leq 2 \wedge w_1 = 1 \wedge w_2 = 2 \wedge a = w_1 \wedge a = w_2 \wedge w_1 \neq w_2$$

この式を先ほどと同様に専用ソルバで判定し, SAT であれば SAT と返し判定を終了する. UNSAT であればまた別の制約条件を試していく. すべての制約条件を試しても UNSAT となった場合は, SAT ソルバに別の真偽値割り当てを探索させ, 判定を続けていく. SAT ソルバが UNSAT と返した場合は, その式は UNSAT となる.

4.3 制約条件の削減

今回の実装で作成する制約条件は，共通変数が増えることで爆発的に増加してしまう. 共通変数の種類を n とすると，作成される制約条件は $2^{(n*(n-1))/2}$ となる．効率化を行う上で，この削減は極めて重要である．

ここで，先ほどの式 φ の場合を考える．式 φ と制約条件は前節で示したが，制約条件 2 をみると w_1, w_2 の間に推移律の矛盾が生じ，制約条件のみで UNSAT となっていることがわかる．このような制約条件はそれ自体が UNSAT であるため，問題式に追加し判定を行っても必ず UNSAT となってしまう．このような制約条件を削除することで制約条件の増加を抑えられないか調査を行った．実際に作成した制約条件に UNSAT となるものがどの程度含まれているのか調べた結果を以下の表 4.3 に示す.

表 1: 制約条件の削減

共通変数	3	4	5	6
制約条件	8	64	1024	32768
充足可能数	5	15	52	203

この結果から共通変数の数が増加すると作成される制約条件の大部分が UNSAT となり，不要なものであることがわかる．また，制約条件が UNSAT となる原因が推移律の矛盾であることもわかる．そこで，制約条件を作成する際に，等号理論用ソルバを途中で組み込むことで不要な制約条件の削減を行うプログラムを実装した.

5. まとめと今後の課題

Nelson-Oppen 法などの背景理論の決定手続きの結合法は, SMT ソルバの主要技術の 1 つである．そのため, 本論文では Nelson-Oppen 法の実装技術である DTC の概要を示した．しかし, ソースコードが公開されている Nelson-Oppen 法を組み込んだ SMT ソルバは存在しない．そこで, 現在 DTC をソースコードが公開されている OpenSMT に実装している．また, DTC では共通変数が増えると作成される制約条件が爆発的に増加してしまうという問題がある．その削減方法として等号理論用ソルバを組み合わせ，不必要なものを排除する方法を実装した．すべての実装が完了次第性能評価等をおこないたいと考えている.

謝辞

本研究は一部，文科省科学研究費補助金（基盤 C：No.22500127）の援助を受けている.

参考文献

[1] Nelson, G. and Oppen, D, C.: Simplification by Cooperating Decision Procedures, ACM Transaction on Prograrnubg Languages and systems, Volume 1, 2, pp.245-257(1979)

[2] 岩沼宏治.: 共通記号を持つ背景理論の決定手続きの結合法とその効率化について (修正版), 信学技報, vol.109, no.456, pp.115-120(2010)

[3] Daniel Kroening, Ofer Strichman.: Decision Procedures An Algorithmic Point of View, Springer-Verlag New York Inc(2010)

[4] 岩沼宏治. 鍋島英知.: SMT:個別理論を取り扱う SAT 技術, 人工知能学会誌, 25 巻 1 号, pp.86-95(2010)

[5] Marco Bozzano, Roberto Bruttomesso, Alessandro Cimatti, Tommi Junttila, Silvio Ranise, Peter van Rossum, Roberto Sebastiani.: Efficient theory combination via boolean search, Information and Computation, Volume 204, pp 1493-1525(2006)