

第19回(最終回)

形式的検証技術の現状と今後の動向

藤田昌宏

この連載では、システムLSIを中心としたハードウェア設計検証を形式的に行う手法について解説してきました。連載の最後に、形式的検証技術の現状と今後の動向について、いくつかの観点から、個人的視点も含めて説明します。(筆者)

論理式の計算機上における取り扱いに関する技術は、近年大きく進歩しています。そしてそれに関連して、形式的検証技術が適用できる範囲が広がりつつあります。ここではその状況、ならびに仕様記述法の動向などを中心に解説します。

現状をごく簡単にまとめると、形式的検証ツールや手法は、誰でも使えるという段階ではありません。依然として「よく分かっている人が賢く使えば、非常に強力なツールである」という事実は変化していません。しかし、技術の

進歩は確実に進んでいること、そして仕様記述言語の標準化が大きく進んだこと、それにツールの使い方に関する解説も増えていることから、以前より、賢く使っている人が増えていることは事実です。いくつかの技術的分野で大きな進歩もあり、システム・レベル設計など、高位設計支援技術の導入が進むであろう今後は楽しみであるといえます。以下では、いくつかの観点から現状と今後について解説します。

1 SAT手法は近年、大きく進歩している

形式的検証に関する基盤技術の中で、近年もっとも研究が進み、結果としてツールの性能が飛躍的に向上しているものに、論理関数の充足可能性(satisfiability)判定手法、いわゆるSAT手法があります。

● SAT問題を解くSATソルバ

SAT問題(論理関数の充足可能性判定問題)とは、図1に示すように、通常は積和形(conjunctive normal form: CNF)で表現された論理式全体を1とするような変数の値の組み合わせが存在するか否かを判定する問題です。問題の難しさの観点からは、NP完全(NP-complete)になります。つまり、論理変数の数に対して、最悪の場合、処理に指数的な時間が必要となると考えられています。

形式的検証手法の多くの問題は、いくつかの変換を行うことにより、SAT問題に変換できることが分かっています。このため、SAT問題と解くツールであるSATソルバ

与えられた和積形論理式(CNF formula) f :

- 変数の集合 V (a, b, c)
- 項(clause)の積 (C_1, C_2, C_3)
- 項: リテラル(literals; 変数がその否定)の和

すべての項の少なくとも一つのリテラルを1とするような変数値の組み合わせが存在するか?

$$\text{例: } \underbrace{(a+b+c)}_{C_1} \underbrace{(\bar{a}+c)}_{C_2} \underbrace{(a+\bar{b}+c)}_{C_3} \quad a=b=c=1$$

図1 SAT(論理関数の充足可能性判定)問題

通常は積和形(conjunctive normal form: CNF)で表現された論理式全体を1とするような変数の値の組み合わせが存在するか否かを判定する問題。

KeyWord

形式的検証, SAT手法, モデル・チェッキング, Bounded解析, 等価性検証, 仕様記述手法, PSL, SVA, 検証IP

は形式的検証手法にとって、非常に重要な要素技術となっています。

● SAT 手法で扱える変数の数が急激に増大

SAT 問題に対する基本的なアルゴリズムとしては、DPLL アルゴリズムが1960年代に考案されました。そしてそれ以降、さまざまな改良がなされてきましたが、1990年代までは、扱える論理変数の数はそれほど大きくならず、結果として形式的検証分野からの興味は、それほど大きくありませんでした。

しかし、図2に示すように、2000年辺りを境として、扱える変数の数が急激に増大しました。図では、横軸が年代であり、縦軸は扱える変数の数を示しています。問題の難しさは、扱える変数の数に対して指数的に増大するはずですが、にもかかわらず、図2のグラフでは、年代に対して扱える変数の数があたかも指数的に増大しているのではないと思われるくらいになっている点に注意する必要があります。これは、計算機科学的な常識ではありえない事が起こっていることになります。

実際には、以下で示すような改良が1990年代後半から行われ始めたためです。これが現在まで続いている結果として、図2に示すような規模のSAT問題を解くことのできるSATソルバが開発されています。

- 基本アルゴリズムは変数の場合分けを繰り返し、途中で矛盾が生じたらほかの場合を総当たり的に調べるということであるが、一度矛盾が生じた場合にその原因を特定し、同じような矛盾に対する解析を二度としないように改良した。
- 変数の値が部分的に決まった際の影響がほかにも伝播することを計算するアルゴリズムを大きく改良した。
- 実際のプログラムとして性能向上を図るために、キャッ

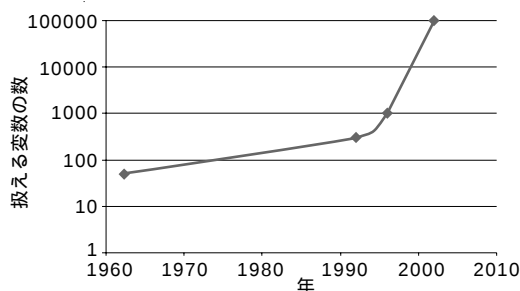


図2 SAT 手法が急速に進歩

2000年辺りを境として、扱える変数の数が急激に増大した。

シュ・ヒット率なども考慮してデータ構造や処理アルゴリズムをチューニングした。

- 変数の場合分け順の決定アルゴリズムにさまざまな工夫を施した。

SAT ソルバに対するコンテストが毎年開催されています。このコンテストでは、前年度の優勝ツールそのままでは次年度に入賞すらできません。SAT ソルバについては、このような速いペースの性能改善が今でも続いています。結果として、現状で、数百万変数からなる論理式の充足可能性判定を数時間で行えるまでになっています。

論理式を計算機で扱うための技術として、2分決定グラフ (BDD: binary decision diagram) があります。しかし BDD では数百変数の論理式を扱うのが限度です。このことを考えると、SAT ソルバの性能は驚異的であると言えます。

● SAT ソルバでは10万ゲート程度まで扱える

順序回路の検証の場合、問題の複雑さという観点からは、回路中のゲート数と順序回路を時間軸方向に何サイクル解析するかということから議論できます。

今、10万ゲートの回路があったとします。それを10サイクル解析する問題をSAT問題に変換すると、(10万×10サイクル×各ゲートの入出力数)だけ変数が現れ、数百万変数が必要になります。現状のSATソルバの性能はそれくらいが限界であるため、その意味で、10万ゲート程度まで扱えることになります。

この規模は、設計者が自分の設計部分を気軽に検証するには少し小さいかもしれません。しかし、設計をいくつかに分ければ、検証できることを意味しています。今後もSATソルバの進歩は続くと考えられることを考慮すると、SAT手法による形式的検証は実用段階に入っていると言えるでしょう。

商用のモデル・チェックング・ツールなどでも、SATソルバが内蔵されているのが普通です。そして、最近の進歩がそのまま取り込まれることも多くなっています。

また、高位設計検証では、扱う変数が整数型など論理変数に分解したくない場合があります。そのような整数変数を含む論理式のある範囲のものは、一定の自動変換によって、SAT問題に帰着する場合も多あります。高位設計検証ツールでは、そのような変換も利用しながら、SATソルバが効果的に利用されています。

なお、最新のソルバも含めて、多くのSATソルバは、パ