

仕様が漸化式で与えられた組合せ回路の形式的設計検証

5B-5

越智裕之 矢島脩三
 京都大学工学部情報工学教室

1 はじめに

形式的設計検証は設計が仕様を満足するかどうかを厳密かつ完全に保証しようというものであり、これまで様々な手法が提案されてきた。特に二分決定グラフ(BDD)が提案されてから飛躍的に進歩し、組合せ回路の論理照合や、順序回路の命題時相論理モデルチェックなどは実用レベルに達しつつある。しかしながら、形式的設計検証では仕様を厳密かつ正確に与えることが必要であり、これは容易ではない。特に組合せ回路の論理照合では、仕様として論理関数を与えるのが一般的であり、同じものを2回設計して比較しているに過ぎないという見方もできる。

本稿では、算術演算回路のような組合せ回路の仕様としてその性質を漸化式で与えることとし、これを設計された回路が満足するかどうか形式的設計検証する方法について考える。漸化式は設計対象の数学的な性質等から容易に導出できることが多いと考えられる。漸化式をBDD上で効率良く取り扱うため、関数 $f(x)$ を表すBDDから関数 $f(x+1 \pmod{2^n})$ を表すBDDを生成する真理値巡回演算のアルゴリズムを示す。

2 準備

本稿では論理変数、即ち0または1の値をとる変数を x_i 、整数変数、即ち整数値をとる変数を x などと表記する。同様に、 f_i 、 f はそれぞれ論理関数、整数関数を表す。

順序付き二分決定グラフ (Ordered Binary-Decision Diagram: OBDD) は、非巡回有向グラフによる論理関数の表現である [1][2] (以下単に二分決定グラフ (BDD) と呼ぶ)。BDD の例を図1に示す。

各非終端ノード (以下単にノードと呼ぶ) は、論理変数がラベル付けられている。各ノードからは、ちょうど2本のエッジが出ており、これらをそれぞれ'0'エッジおよび'1'エッジと呼ぶ。BDDの各パスには同じ変数のノードは2回以上出現しない。また、あるBDD中の全てのパスで、変数はある一定の順序で出現する。

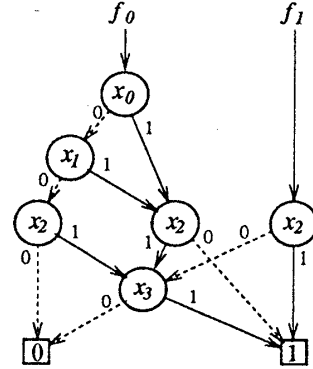


図1: 二分決定グラフ (BDD) の例

BDDには、(1) 表現が一意なので論理関数の等価性判定が容易である、(2) 実用的な論理関数の多くのものが比較的少ないノード数で表せる、(3) 論理関数同士の論理演算がグラフのノード数に比例する時間でできる、などの特長がある。

3 提案手法

3.1 仕様

例として、自乗関数 $f(x) = x^2 (0 \leq x < 2^n)$ を計算する組合せ回路を考えよう。 x は n ビットの2進数で符号化され、 n 個の論理変数 $\{x_0, \dots, x_{n-1}\}$ で与えられるとする (LSB は x_0)。同様に、 f も $2n$ ビットの2進数で符号化され、多出力論理関数 $\{f_0, \dots, f_{2n-1}\}$ で実現されるとする (LSB は f_0)。

上の自乗関数の仕様は、以下のように表すことができる。これが自乗関数の漸化式による仕様記述である。

$$f(0) = 0$$

$$f(x+1) = f(x) + 2x + 1 \quad (0 \leq x < 2^n - 1)$$

以下、設計された組合せ回路が上記の仕様を満たすことを検証する方法について考える。

3.2 検証手法

提案する検証アルゴリズムを自乗関数の検証を例にとって書くと以下ようになる。

Formal Design Verification of Combinational Circuits Specified by Recurrence Equations
 Hiroyuki OCHI and Shuzo YAJIMA
 Department of Information Science, Faculty of Engineering,
 Kyoto University, Kyoto, 606-01, JAPAN

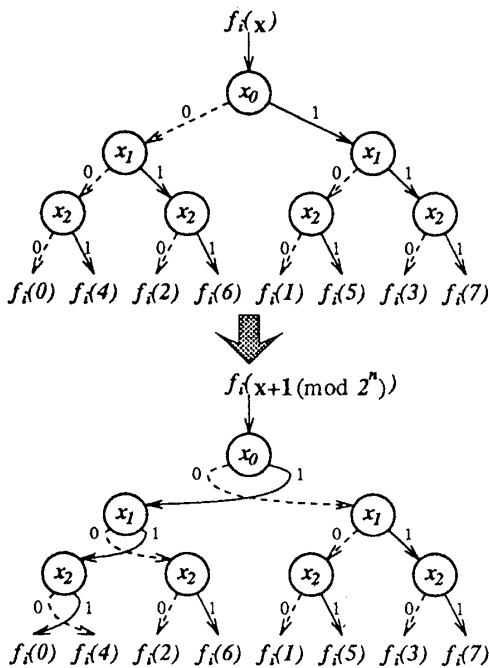


図 2: BDD 上での真理値巡回演算

1. 設計された組合せ回路の記述から、それを表す BDD を生成する。これが実現する算術関数を $\hat{f}(x)$ とする。
2. $\hat{f}(0) = 0$ を確認する。
3. $\hat{f}(x)$ を表す BDD をもとに $\hat{f}(x+1)$ を表す BDD を構成する。
4. $\hat{f}(x)$ を表す BDD をもとに $\hat{f}(x) + 2x + 1$ を表す BDD を構成する。
5. 3. と 4. が一致することを確認する。

1. は従来の記号シミュレーションと同様に、回路記述中のゲートに対応する論理演算を BDD 上で行なうことによりグラフを構成すればよい。2. は BDD の根から 0 エッジをたどって到達する葉を調べればよい。4. は全加算器をシミュレートすることにより、1. と同様に構成することができる。5. は論理関数の一致判定であり、BDD 上では容易である。(厳密には $x = 2^n - 1$ の場合を除外しなければならないが、以下に述べる方法では自乗器の場合その必要がない。)

3. は、BDD の変数が根に近い方が LSB、葉に近い方が MSB と順序づけられている時、図 2 のように $2n$ 本のエッジをつなぎ替えた BDD を構成することによりほぼ実現できる。この操作を真理値巡回演算と呼ぶ。

4 実現と評価

3 章で提案した手法に基づく自乗器の形式的設計検証プログラムを SPARC station 10 model 51 上を実現し、実験を行なった。湊氏の作成した共有二分決定グラフパッケージ [3] を利用し、検証の対象としては Wallace 木乗算器の乗数と被乗数に同じものを入力する回路を用いた。

例えば 20bit 自乗器の場合、1. のための CPU 時間が 32.71 秒、2. ~ 5. のための CPU 時間が 10.00 秒であった。また、1. で得られたグラフのノード数、及びこれと 3. で得られたグラフを共有した場合のノード数は、それぞれ 146,781、146,985 であった。これより、検証対象の回路を BDD で表現できてからの検証は極めて高速に行なわれることがわかる。

5 おわりに

本稿では、算術演算回路のような組合せ回路の形式的設計検証にあたり、その仕様としてその性質を漸化式で与えることを提案し、次に、これを実現するための効率の良い検証手法を提案した。さらに実験結果より、これが実用化可能であることを示した。

形式的設計検証が幅広く利用されるためには、これからも仕様の記述方法について研究されなければならない。本稿で提案した仕様の記述方法及び真理値巡回演算は [4] と組み合わせることによりさらにその有用性が高まると期待される。また、真理値巡回演算が他の分野に応用できないか、興味深い。

なお、本研究の一部は文部省科学研究費補助金 (特別研究員奨励費) による。

参考文献

- [1] S. B. Akers: "Binary Decision Diagrams", IEEE Trans. Comput., vol. C-27, no. 6, pp. 509-516, (June 1978).
- [2] R. E. Bryant: "Graph-Based Algorithms for Boolean Function Manipulation", IEEE Trans. Comput., vol. C-35, no. 8, pp. 677-691, (Aug. 1985).
- [3] S. Minato, N. Ishiura and S. Yajima: "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation", Proc. 27th ACM/IEEE DAC, pp. 52-57, (June 1990).
- [4] Y. T. Lai and S. Sastry: "Edge-Valued Binary Decision Diagrams for Multi-Level Hierarchical Verification", Proc. 29th ACM/IEEE DAC, pp. 608-613, (June 1992).