

第4回

フォーマル・ベリフィケーションの基本原則(3) ——2分決定グラフの変数順序を決める

藤田昌宏

今回は、論理関数の正規表現である2分決定グラフの変数順序の決めかたについて解説する。同じ論理関数を表現した2分決定グラフであっても、変数順が変わるとノード数が大きく変化する。コンピュータ上で高速に処理するためには、ノード数が少なくなる(つまり、計算量やメモリ使用量が減る)変数順序を見つける必要がある。ただし、乗算器のように、変数順をどう変えても、2分決定グラフのサイズが爆発するタイプの回路もある。(編集部)

前回まで、フォーマル・ベリフィケーション(形式的検証)を支える基礎技術として、計算機上で使われる論理関数のコンパクトな表現法である「2分決定グラフ(BDD: binary decision diagram)」について説明してきました。今回は、2分決定グラフを利用するうえで重要な「変数順の決定・最適化手法」について説明します。前回も述べたように、2分決定グラフの大きさは変数順によって大きく変化するのが普通です。論理関数の中には変数順の影響を受けない(変数順最適化の効果があまりない)ものも多いのですが、加算器を表す論理関数や制御回路の論理など、LSIの実設計で利用される論理関数については変数順に依存することが少なくありません。今回は、2分決定グラフのための変数順決定法や変数順の逐次改善手法について説明したいと思います。

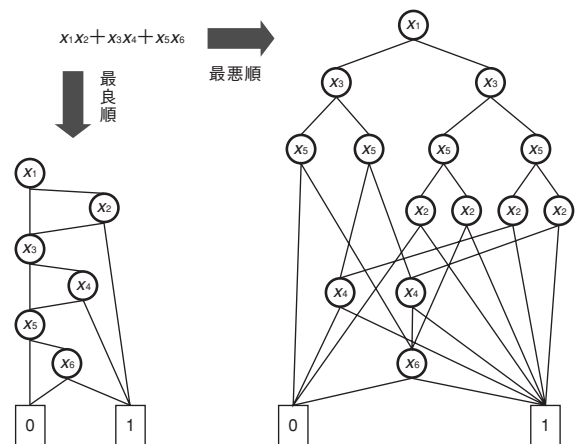
●よく考えられた回路図の変数順はBDDを小さくする

2分決定グラフの大きさは、使用する変数順に大きく依存します。前回も示しましたが、例えば図1の論理関数 $x_1x_2 + x_3x_4 + x_5x_6$ に対する2分決定グラフを作成するとき、変数順が「 $x_1, x_2, x_3, x_4, x_5, x_6$ 」か「 $x_1, x_3, x_5, x_2, x_4, x_6$ 」かによって、必要なノード数が異なります。最良変数

順の一つである「 $x_1, x_2, x_3, x_4, x_5, x_6$ 」の場合、ノード数は変数の数に比例します。一方、最悪変数順の一つである「 $x_1, x_3, x_5, x_2, x_4, x_6$ 」の場合、ノード数は変数の数の指数に比例します。

このように、2分決定グラフを利用する立場からすると、変数順の決めかたはきわめて重要な問題となります。この変数順は、EDAツール(フォーマル・ベリフィケーション・ツールなど)によって自動的に決められることが多いのですが、場合によってはユーザが手を加えることもできるようです。

LSIの回路において、論理関数全体の値を決めるとき、一般に、制御部の変数の値はデータパス部の変数の値より影響力が大きいと考えられます(制御部のほうが、データパス部より結線の数が多く、回路構成が複雑になるケース



【図1】変数順による2分決定グラフの大きさの変化

2分決定グラフの大きさは、変数順に大きく依存する。同じ論理回路の出力に対する2分決定グラフでも、最良の変数順と最悪の変数順ではノード数が大きく異なる。この例では、最良変数順の場合は回路の入力数に比例するノード数になるが、最悪順の場合は回路の入力数の指数に比例するノード数になってしまう。

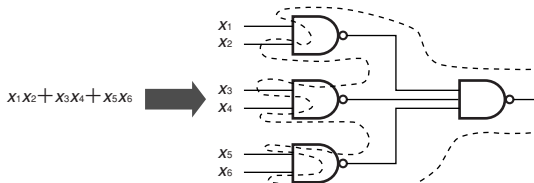
が多いため)。そのため、2分決定グラフの変数順を決めるときに、最初に制御部の変数を並べて、その後にデータパス部の変数を並べるといった方法がよく用いられます。あるいは、後に述べるように、「同じゲートに接続されている信号間の変数順は近いほうがよい」という規則に基づいて決めることも考えられます。

人手で変数順を決める場合の一つの方法は、回路図上の変数順をそのまま2分決定グラフの変数順として採用することです。よく考えて作成された回路図の場合、同じゲートに接続される信号は近くに集められています。また、見やすくするため、できるだけ信号線の交差が少なくなるように描かれています。このように「よく考えて作成された回路図」が存在する場合には、回路図上に現れる変数順をそのまま2分決定グラフの変数順にすると、比較的良好な結果が得られることが経験的にわかっています。

現在のLSI設計では論理合成ツールなどを利用することが多く、最近では設計者が回路図を作成することはまれです。そのため、実際にはこの方法が使えないかもしれませんが、知っておいても損はないでしょう。

●静的な決定手法と動的な改善手法がある

実際のLSI設計では、設計者自身が2分決定グラフの変数順を指定することはめんどくさく、かつ手間や時間もかかるため、ほとんどの場合、変数順はツール側で自動的に決定されています。2分決定グラフの変数順を決める方法には、2分決定グラフを作成する前に決定する「静的な変数順決定手法」と、いったん2分決定グラフを生成してから（あるいは生成する途中で）変数順を改善する「動的な変数順改善手法」の2種類があります。ここでは、まず、静的な



【図2】回路を出力から入力へ向かって深さ優先でたどる

木構造となっている回路、つまり、すべてのゲートの出力が一つのゲートにしかつながっていない回路（再取れのない回路）の場合には、回路のネットリストを出力から入力方向へ深さ優先でたどっていき、現れた外部入力の順番を2分決定グラフの変数順とすることで、最適な変数順が得られる。これは単純な深さ優先なので、2段目のNANDゲートの入力をどの順にたどるかは自由であり、その意味で、いろいろな変数順が得られる。例えば、「 $x_1, x_2, x_3, x_4, x_5, x_6$ 」や「 $x_3, x_4, x_1, x_2, x_5, x_6$ 」、「 $x_5, x_6, x_3, x_4, x_1, x_2$ 」なども最適な変数順となっている。

変数順決定手法の典型的な例を説明し、その後で動的な変数順改善手法を説明します。

実はこれ以外にもさまざまな手法が提案されています。「最適な変数順決定問題」はNP完全問題（NP-complete problem；入力データ量に対して、指数関数的な処理時間がかかってしまう問題）であることが証明されているので、すべての解をしらみつぶしに探索するようなことを行わない限り、最適な変数順を見つけることはできません。そこで、計算時間が少なくすむ各種のヒューリスティック（最適解ではないが、比較的良好な解が得られる経験的手法）が提案されているわけです。ヒューリスティックであるため、うまくいく場合もあれば、うまくいかない場合もあるわけですが、実際の設計に出てくる論理関数については、比較的うまくいくケースが多いようです。

●静的な手法：出力から入力方向へ深さ優先でたどる

静的な変数順を決定するうえで、何が重要かを考えてみましょう。まず、図1の最適変数順がどのような順番になっているかを考えてみます。図1の論理関数は $x_1x_2 + x_3x_4 + x_5x_6$ ですが、これを回路図で書くと図2のようになります。このようにきれいな回路図が描ける場合には、前述のように、回路図に出てくる変数を上から（あるいは下から）順に並べたものを変数順とすれば、結果的に良い2分決定グラフが得られます。

さて、この回路（あるいはネットリスト）から、良い変数順を自動的に得るにはどうすればよいのでしょうか？ 一つ言えることは、同じゲートにつながっている信号は、変数順として、隣り合っているほうがよいということです。これをいちばん簡単に満足する方法は、回路のネットリストを出力から入力方向へ深さ優先でたどっていき、現れた外部入力の順番を2分決定グラフの変数順とすることです。この過程を図2に示します。これは単純な深さ優先なので、2段目のNANDゲートの入力をどの順にたどるかは自由であり、その意味で、いろいろな変数順が得られます。例えば、「 $x_1, x_2, x_3, x_4, x_5, x_6$ 」や「 $x_3, x_4, x_1, x_2, x_5, x_6$ 」、「 $x_5, x_6, x_3, x_4, x_1, x_2$ 」などが考えられます。ここで注意してもらいたいのは、これらのどれもが「最小の2分決定グラフ」を与える変数順だということです。

実用的な回路の多くは、図2に示すような木構造の回路ではなく、一つのゲートの出力が複数のゲートの入力になり、それらの出力が回路の後のほうで再度同じゲートの入