素子制限のある論理回路を等価変換するための基本操作集合について

8M−9

日野 健介 岩間 一雄 澤田 直
九州大学工学部

1. はじめに

論理回路の自動設計は近年その重要性を増々大きなものにしている．数多くの手法や考え方が提案されているが，その代表的なものは，まず何らかの方法で初期回路を設計し，段階的により性質の良い回路に変形していく[3][4]というものである．また，最近注目を集めているアルゴリズムの効率を実験的に評価するためのテスト例題の生成[5]では，逆に性質の良い回路(アルゴリズムが導き出すべき答)から悪い回路(アルゴリズムの入力とする)に変形できる必要がある．いずれの場合も，論理回路の等価変換(回路が実現する論理関数を変えることなく変形する)が鍵になる．

我々はこれまでの研究[6]で，論理回路としては，AND,OR,NOT素子が使える組合せ回路を対象として，任意の論理回路から任意の等価な論理回路へ変換するための完全な基本操作規則の集合を提案した．ここで，“変換”は1回の操作が多項式時間で実行できる“基本操作”の列で定義している．本稿では，使える素子をNANDのみに制限した組合せ回路を対象とする．NANDをANDとNOTの組合せとみれば，NANDに制限された回路は素子制限のない通常のAND,OR,NOTの回路の部分集合とみなせる．従って，上記[6]の規則を適用すれば，任意のNAND回路f_1から任意のNAND回路f_2へ変換することは可能である．しかし，この場合，f_1からf_2への変換の過程で，NAND素子のみでない回路を経由しなくてはならない．本稿の立場は，常にNAND回路という制限を犯すことなく変換を進めていることに注意されたい．いわゆる変換公式についても，AND,OR,NOTの場合に比べて，あまり知られてないようにみえる．NANDという論理はゲートの重要な電子回路実現に本質的に関係しているため，一般にシステムの回路はNAND(あるいはNOR)ゲートを使用して設計されることが多く，実用的な面でも重要である．

2. 回路の定義

2.1 基本方針

我々は回路を以下のような式の集合で表現する．

$g[01] = \langle g[010], \langle g[011], x_3\rangle\rangle$

$g[010] = \langle g[011]\rangle$

$g[011] = \langle x_1, x_2\rangle$

すなわち，回路をいくつかの部分回路に分けて表わす．ただし，ファンアウト数が1の木状回路の場合は，これら部分回路は必要なく，単一の式で表現することも可能である．以下に回路のより厳密な定義を与える．

2.2 部分式の定義

まず，部分式を以下のように$\{0,\ 1,\ 0,\ 1,\ x,\ g,\ [,\ ],\ \langle,\ \rangle\}$上の文字列として定義する．

(1) 0及び1は部分式

(2) $x[a]$は部分式 (aは二進数)

(3) $g[\ell]$は部分式 (ℓは0と1の文字列)

(4) $S_1, S_2, \cdots, S_m (m \geq 1)$が部分式ならば，
$\langle S_1, S_2, \cdots, S_m\rangle$は部分式

ここで，**0** (**1**)は，論理の偽(真)を，0と1は(2)と(3)のaとℓをそれぞれ表現する．$x[a]$は入力変数を表す．(以下，簡単のため$x[a]$をx_i(ただしiはaを十進数に読みかえたもの)と表記することがある)．$\langle S_1, S_2, \cdots, S_m\rangle$は，$m$入力のNANDゲートを表わす．また$g[\ell]$をラベルとよぶ．ラベル$g[a_1], g[a_2]$があるとき，それらの大小関係を次のように定義する．a_1, a_2は0と1の文字列であるが，辞書順にa_1が優先しているとき，$a_1 > a_2$とする(ただし$0 > 1$とする)．つまり最大のラベルは$g[0]$である．

我々は，集合$\{W_1, W_2, \cdots, W_m\}$を式と呼ぶ．ここで，$W_j$は“$g[\ell] =$ 部分式”の形の文字列である．以下，この“部分式”をラベル$g[\ell]$の部分式(の定義)とよぶことがある．

2.3 回路の定義

上記で定義した式が次の条件を満たすとき，回路とよぶ．

(i) ラベル$g[0]$の部分式が存在する．

(ii) ラベル$g[a_1]$の部分式に$g[a_2]$が現れるなら，$a_1 > a_2$である

(iii) ある部分式の中に$g[\ell]$が現れるとき，ラベル$g[\ell]$の部分式の定義が必ず存在する．

(iv) 特定のラベル$g[\ell]$の部分式が2つ以上存在しない．

(v) 式に現れる入力変数x_iは，x_1から始まり，x_2，$x_3, \cdots$のように途切れることなくx_nまで続く．このときn変数の回路であるという．

(i)は，全体の回路を表現する部分式に対してラベル$g[0]$を割り当てるという約束を表す．(ii)は，我々が扱う回路は組合せ回路であるので，回路にフィードバックを許さないための条件である．(iii)では，ゲートへの入力線のすべてに，必ず入力があることを保障している．また(iv)は，一つの入力線に対して複数の入力を許さないというものである．なお，式が回路を表しているとき，式の各部分式を部分回路とよぶこともある．

以上が本稿における回路の定義である．次節ではこれらの回路に対する基本操作について議論する．

3. 基本操作

基本操作は，一部の例外を除き$g \Longrightarrow h$の形で表され，有限個の操作規則の集合の中から1つを選んで，それを適用することによって行なわれる．回路f_1のある部分列gに操作規則を適用しようとするとき，gが規則の左辺とパターンマッチすれば(gのさらに一部分が部分式になっているかどうかの判定も含む)，gをhと置き換えた回路f_2に変換される．これをf_1からf_2への1回の基本変換とよび，より形式的には，関数

$$T(f_1, r, k) = f_2$$

で表す．ここで，f_1は基本操作を行なおうとしている(元の)回路，rは操作規則，f_2は基本操作によって変形された(結果の)回路である．一般にf_1とrを指定しただけでは，適用できる部分列として様々な可能性があってf_2が一意に定まらない．そこでパラメータk(整数)を与える．これによって可能なすべてのf_2の中から適当な順序づけによるk番目を指定する．

本論文で提案する操作規則集合(公理系)は，以下のようなものである．

公理系

(1) $\langle \mathbf{1}\rangle \Longleftrightarrow \mathbf{0}$ $\langle \mathbf{0}\rangle \Longleftrightarrow \mathbf{1}$

(2) $\langle x, x\rangle \Longleftrightarrow \langle x\rangle$

(3) $\langle x, \langle x\rangle\rangle \Longleftrightarrow \mathbf{1}$

(4) $\langle x, \langle\langle y, z\rangle\rangle \Longleftrightarrow \langle x, y, z$

A complete set of basic operations to transform between equivalent switching circuits of NAND gates
Kensuke HINO, Kazuo IWAMA, Sunao SAWADA
Kyushu University

(5) $x, y \Longleftrightarrow y, x$
(6) $\langle x, 1\rangle \Longleftrightarrow \langle x\rangle$
(7) $\langle\langle x\rangle\rangle \Longleftrightarrow x$
(8) $\langle x, \langle y, z\rangle\rangle \Longleftrightarrow \langle\langle\langle x, \langle y\rangle\rangle, \langle x, \langle z\rangle\rangle\rangle\rangle$
(9) $g[\ell] = f$ という部分式の定義が存在するとき,
$g[\ell] \Longleftrightarrow f$
10) ラベル $g[\ell]$ がいかなる部分式にも現れないとき，ラベル $g[\ell]$ の部分式の定義を，回路を表現する部分式の集合から除去できる．これを消去とよぶ．
11) ラベル $g[\ell]$ の部分式の定義が存在しないとき，$g[\ell]$ = 部分式の形の定義を集合に付加できる．ただしそのサイズ (長さ) は，現在の式のサイズの多項式で制限される．これを生成とよぶ．

上記の公理系において，各 x, y, z は，2.節の条件を満たす任意の部分回路にマッチする．また $\Longleftrightarrow$ は左辺から右辺へ及び右辺から左辺への両方向の操作が可能であることを意味する．なお，消去と生成は互いに逆操作の関係にあることに注意されたい．

公理 (9) は，いわゆる代入とその逆の操作を表現している．ある部分式にラベル $g[\ell]$ が現れているとき，その $g[\ell]$ を $g[\ell]$ の定義によって置き換えること，またその逆として，ある部分式中の式 (文字列)σ に対して，$g[\ell] = \sigma$ という定義が存在する場合は σ を $g[\ell]$ と置き換えることができる．$g[\ell] = \sigma$ の定義が存在しない場合，公理 (11) の生成によってそれを作り出してから公理 (9) を適用する点に注意されたい．

これら基本操作規則の組合せにより，例えば次のような操作規則を導くことができる．

公理 (4) の一般化

$$\langle X_1, \cdots, X_l, \langle\langle S_1, \cdots, S_m\rangle\rangle, Y_1, \cdots, Y_n\rangle \Longleftrightarrow \langle X_1, \cdots, X_l, S_1, \cdots, S_m, Y_1, \cdots, Y_n\rangle$$

公理 (8) の一般化

$$\langle X_1, \cdots, X_l, \langle S_1, \cdots, S_m\rangle, Y_1, \cdots, Y_n\rangle \Longleftrightarrow \langle\langle\ \langle X_1, \cdots, X_l, \langle S_1,\rangle, Y_1, \cdots, Y_n\rangle, \cdots \cdots, \langle X_1, \cdots, X_l, \langle S_m\rangle, Y_1, \cdots, Y_n\rangle\ \rangle\rangle$$

次に基本操作に要する時間についての定理を示す．

[定理 1] 1回の基本操作は，多項式時間で実行できる．

(証明) 例えば，回路 f_1 に操作規則 (8) $\langle x, \langle\langle y, z\rangle\rangle \Longrightarrow \langle x, y, z$ を適用する場合に考慮する点は，部分列 $g = \langle x, \langle\langle y, z\rangle\rangle$ としての全ての可能性，g の中で x, y, z としての全ての可能性，x, y, z のそれぞれが式かどうかの判定の 3 点であるが．それぞれが多項式時間で実行できることは容易に判ろう．よってこの操作は多項式時間で実行できる．公理 (1)〜(10) に関しては全く同様である．公理 (11) については，生成できる部分式の長さに制限を設けている点に注意されたい． □

4. 基本操作規則集合の完全性

3.節で与えられた公理系によって定義される基本操作規則集合によって，任意の等価な回路 f_1 と f_2 の間で変換が可能であることを示す．ここで変換が可能であるとは，f_1 から f_2 へ変換する基本操作の列が存在することを意味する．(そのような列は一般に一意ではない．例えば，短い列をどのようにして得るかは難しい問題である．) 基本方針は 3.節の基本操作規則集合によって論理式を「標準形」に変換できることを示すことである．標準形は特定の論理関数に対しては唯一つしか存在しないように決める．回路 f_1 と f_2 が等価であるとする．このとき共に等しい標準形 p へ f_1 からは基本変換列 σ_1 で，f_2 からは基本変換列 σ_2 で変換できるとする (定理 2)．ここで，3.節の規則が両方向になっていることに注意されたい．つまり σ_2 を逆方向に適用することにより p から f_2 へ変換できる．よって f_1 から f_2 への変換が可能であることが示せる．

4.1 標準形

標準形は特定の論理関数に対しては唯一つしか存在しないようにするため，以下のように定義する．n 変数の回路の標準形は $\langle p_1, p_2, \cdots \cdots, p_j, p_{j+1}, \cdots, p_m\rangle$ (特別の場合として 0 を含む) の形である．ここで，各 p_j は $\langle X_1, X_2, \cdots, X_n\rangle$ の形をしており，X_i は変数 x_i またはその否定 $\langle x_i\rangle$ である．任意の $j \geq 1$ に対し，p_j と p_{j+1} の間には以下の関係がなければならない．$N(p_i)$ を $X_1, \cdots, X_n$ の各 X_i に対しそれが x_i なら 1，$\langle x_i\rangle$ なら 0 に置き換えて得られる二進数とする．このとき $N(p_j) < N(p_{j+1})$ である．

4.2 主定理

以下に，本稿の主定理を示す．

[定理 2] f_1 が回路ならば f_1 を標準形に変形する基本操作の列が存在する．

(証明) 回路 f_1 を標準形に変換するアルゴリズムが存在する (4.3節)．このアルゴリズムは，回路 f_1 を文字列 f として入力すると，標準形を生成して終了する．ここで適用する操作は，基本操作規則またはその組合せにより導かれる操作である． □

4.3 標準形への変形

任意の回路を標準形に変形する方法の概略を以下に示す．

Step1：回路を木状にする．
Step2：回路を 3 段以下にする．
Step3：定数の除去 及び 重複したリテラルのゲートへの入力を 1 つにする．
Step4：回路を 3 段にする．
Step5：含まれないリテラルの追加．
Step6：変数及びゲートの順番を並べ換える．

上記のアルゴリズムで行なう操作の中には 基本操作規則にはないものも含んでいる．これらは，アルゴリズムの記述を容易にするためのものであって，基本操作規則から必ず導くことができる．

5. おわりに

本稿では，NAND 素子に制限された“回路”を部分式の集合として定義し，1 回の操作が多項式時間で実行できる基本操作規則の集合を提案し，その基本操作規則集合が任意の回路から任意の等価な回路への変換に対して完全であることを示した．今後は，本研究をもとに目的の一つであるアルゴリズム評価のためのテスト例題生成への応用について考えていく．

参考文献

[1] 室賀, 笹尾, “論理設計とスイッチング理論”, 共立出版, (1979).
[2] Zvi Kohavi, “Switching and finite automata theory”, McGraw-Hill, Inc. (1978).
[3] R.K.Brayton, et al. “MIS: Multi-level Interactive Logic Optimization System”, *Trans. on CAD*, Vol.CAD-6, pp.1062-1081, Nov. 1987.
[4] S.Sawada, et al. “Generation of Fan-in Restricted Initial Networks for Transduction Method”, *Proc. the Synthesis and Simulation Meeting and International Interchange (SASIMI '92)*, Kobe, Apr. 1992.
[5] K.Iwama, H.Abeta and E.Miyano, “Random Generations of Satisfiable and Unsatisfiable CNF Predicates”, Proc. IFIP 12th World Computer Congress, pp.322-328(1992)
[6] 日野, 岩間, 澤田, “組合せ論理回路を等価変換するための基本操作集合について”, 平成 4 年 11 月コンピュテーション研究会.