パストランジスタ回路の網羅的列挙による素子数最小回路の探索

清水亮太[1,a)]　湊 真一[1,b)]

概要：パストランジスタ回路のような導通/非導通のスイッチを自由に接続し作成される論理回路を変数ラベル付きのグラフ構造でモデル化した際に，指定したサイズ以下のグラフで設計可能な全ての論理関数を網羅的に列挙する方法について述べる．ある論理関数を最も少ない素子数で表現するパストランジスタ回路を探索する方法や，パストランジスタ回路になるためのグラフ的性質や論理関数の同値類と呼ばれる特徴を用いてアルゴリズムの高速化を図る手法などの研究内容を示す．また，本手法を適用することで列挙されたグラフを用いて，3 入力までの論理関数を表現する最小のパストランジスタ回路を明らかにする．

Exploring Minimum Pass Transistor Circuits Based on Exhaustive Enumeration

Ryota Shimizu[1,a)]　Shin-ichi Minato[1,b)]

Abstract: We model pass-transistor circuits with the graph whose edges are labeled with input variables, and we show a method of exhaustive enumeration of all Boolean functions designed by the graphs bounded by a given size. We also show a method for finding the minimum pass-transistor circuit that represents a given Boolean function, and discuss the way to speed up the enumeration algorithm based on some properties of graphs and NP-equivalent class of Boolean functions. Finally, we show a catalogue of the minimum pass-transistor circuits that represent all three-input Boolean functions.

1. はじめに

集積回路の発展は近年の情報技術の進歩に大きく貢献しており，様々な回路や設計方式が検討されている．そのうちの一つにパストランジスタ回路 [1], [2], [3] が存在する．ノード間を直接トランジスタで結び，スイッチとして動作させるパストランジスタ回路は省電力化や省面積化などを目的として幅広く利用されており，BDD[4] を用いた設計を元に基本セルが作成される例 [2] など様々に BDD への適用や基本セルが提案されてきた．また，近年では光スイッチ回路など先端デバイスにおいてスイッチを組み合わせた回路も検討されるなど応用も期待されている．

一方，列挙とは与えられた条件を満たす解を全て出力することであり，従来から幅広い分野で利用されている．回路分野では 3 入力論理関数を表現する最小の NAND/NOR 回路の列挙 [5]，グラフに対しては非同型グラフの列挙問題 [6] や二端子直並列グラフの列挙 [7] の例などがあるが、我々の知る限りにおいてすべての論理関数に対して最小パストランジスタ回路を列挙したという研究は見当たらなかった。

我々は網羅的なパストランジスタ回路の全列挙という手法を用いて，ある辺数以下のパストランジスタ回路を列挙し，その表現する論理関数を確認する方法について研究を行った．2 章ではパストランジスタ回路の列挙の前提や定義を確認し，3 章では列挙のアルゴリズムを順に述べる．4 章では実際に実装を行い，3 入力以下の論理関数を表す最小素子数のパストランジスタ回路について明らかにした結果を示し，5 章では結論を述べることとする．

1　京都大学 大学院 情報学研究科
a)　shimizu.ryota.84e@st.kyoto-u.ac.jp
b)　minato@i.kyoto-u.ac.jp

ⓒ 2020 Information Processing Society of Japan

2. 準備

2.1 パストランジスタ回路のモデル化

パストランジスタ回路 (図 1) をそのまま扱うと煩雑となるため，グラフ構造を用いてモデル化を行う．我々はパストランジスタ回路の二つの要素に着目しモデル化を行う．まず一つ目の要素はパストランジスタの接続の仕方である．パストランジスタの結節点を頂点とし，パストランジスタを辺とする無向グラフを考える．その際，パストランジスタ回路の二端点を特別な頂点と見なしモデル化を行い，2 つの頂点が特別に区別されているグラフを二端子グラフ (図 3) と定義する．また，本稿内では特別な 2 頂点を S と T と名付けそれ以外の頂点を数字で表記する．二つ目の要素は各トランジスタのゲートに入力されている変数である．それについては，二端子グラフの辺に変数のラベルを付与することで表現する．そのようなグラフを変数ラベル付き二端子グラフ (図 2) と呼ぶこととする．

2.2 二端子グラフとその性質

二端子グラフが他の頂点と区別された二頂点を持つ無向グラフであることは先の節で述べた．また，今回考える二端子グラフはパストランジスタ回路をモデル化したものであるので，基本的にグラフは連結なものを考える．また，両端点 ST を区別することなく，同型性を考えることとする．同型性に関する詳しい定義は 3.3.3 節で述べる．それらに加え，パストランジスタ回路からモデル化された二端子グラフは更なる特徴を有している．それは全ての辺が S と T を結ぶ単純パスの一部となっているということである．図 4 のようにどの ST パスにも寄与しない辺が存在するとき[*1]，このような辺を取り除いても表現される論理関数は変化しない．そのため，列挙する二端子グラフは「全ての辺が ST パスの一部となっているもの」を考えることとする．この性質を非冗長性と定義し，そのような性質を持つ二端子グラフのことを非冗長二端子グラフと呼ぶ．

3. パストランジスタ回路の網羅的列挙

3.1 問題設定

本研究では先の条件を満たす二端子グラフに対して，次の二つの問題に対して検討を行うことを目的として列挙アルゴリズムを作成する．

- ある論理関数を表現する変数ラベル付き二端子グラフは何辺からなるものが最小か？
- ある二端子グラフに異なる変数割り当てを与えた際に，表現できる論理関数の種類は何種類か？

前者はある論理関数を表すことのできる最小辺数の二端子グラフを明らかにする問題であり，後者はどのような二端子グラフが多くの関数を表現可能かという問題である．本稿では辺が少ない二端子グラフから順に表現できる論理関数の確認を行い，それらの結果を纏めることで前者後者双方の問題を検討する．

[*1] 単純パスとは同じ頂点が複数回出現しないパスである。頂点 1-2,2-3,3-1 間の辺を通ろうとすると頂点 1 を 2 回通ることとなるため、単純パスではなくなる

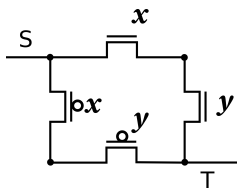


図 1 パストランジスタ回路

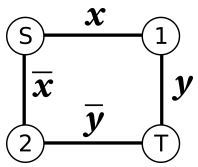


図 2 変数ラベル付二端子グラフ

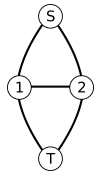


図 3 二端子グラフ

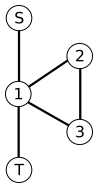


図 4 冗長な二端子グラフ

3.2 全列挙のための手続き

全列挙を行うにあたり段階を踏んで手続きに沿って処理を行う．手続きの順番を以下に示し、手続きの概略図を図 5 に示す。

(1) 単純な二端子グラフを列挙する．
(2) 作成された二端子グラフに対して，多重辺を付与する．
(3) 多重辺が許容された二端子グラフに対して，各辺に変数ラベルを割り当てる．
(4) グラフの形状と割り当てられた変数ラベルから，表現する論理関数を確認する．

具体的な詳細は 3.3 節から 3.6 節で一つずつ述べるがそれぞれについて概説する．手順 1 はまず変数ラベルを割り当てる前の単純二端子グラフそのものを作成する手順である．単純であるとは，多重辺が存在しないことを指す．非冗長性を確認したり，同型のものを取り除くこともこの段階で行う．概略図中の左端のグラフと右端のグラフは同型となっているため右端のグラフは取り除かれる。手順 2 では，先ほど生成した単純二端子グラフに対して多重辺を付与し，非同型な多重辺を許す二端子グラフを列挙する．その目的等は後述する．手順 3 では多重辺を許した二端子グラフに対して，それぞれの辺に変数ラベルを付与する．その際 NP 同値類の考え方を利用し高速化を行うため，その手法についても説明を行う．手順 4 ではその確認の方法について簡単に述べる．

3.3 単純な二端子グラフの列挙

今回単純な二端子グラフ列挙のアルゴリズムにおいて，

ⓒ 2020 Information Processing Society of Japan

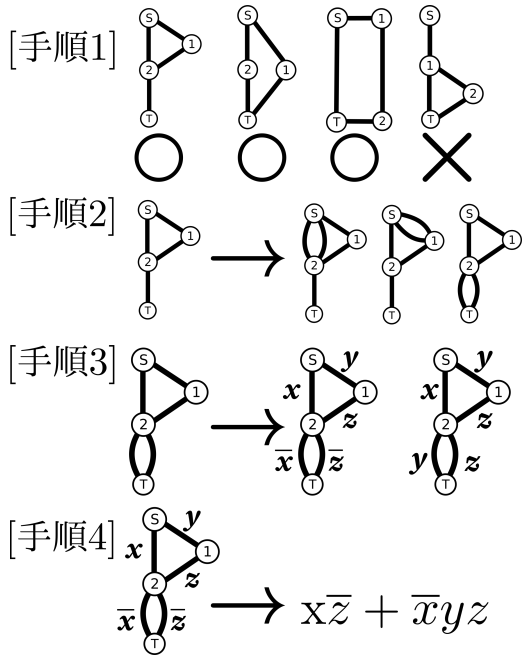


図 5　今回のアルゴリズムの概略図

入力は生成されるグラフの頂点数と辺数であり，出力は単純で非同型非冗長な二端子グラフの集合である．非同型で非冗長なものに限定する目的としては，重複しているものや無駄があるものを取り除き，以降の処理を高速に行うためである．この生成アルゴリズムは三つの段階に分けられる．1 つ目は出力されるグラフの候補グラフを列挙する段階である．この段階のグラフには冗長なものや同型なものが含まれるが，漏れがないように列挙することに注意する．二つ目の段階は，冗長性を判定することで非冗長なもののみを残す処理である．最後に 3 つ目の段階で同型性判定を行うことで，目的の出力を得ることとなる．

3.3.1　候補グラフの生成

この段階では頂点数と辺数を入力とし，候補グラフを出力する．頂点 S,T および番号付けされたその他の頂点について，深さ優先探索を用いて各頂点間に辺を張っていき，指定した本数が張られた段階で出力を行う．この処理は枝狩りを行うことで高速化できる．具体的には，S と T の頂点の次数は 1 以上かつその他の頂点は次数が 2 以上となることが，非冗長性および二端子グラフの成立を満たす必要条件となっているため，これ以上残りの辺を追加しても上記の制約を満たすことができなくなった時点で枝狩りを行うことができる．

3.3.2　非冗長性判定

2.2 節で二端子グラフが非冗長であることの定義は，全ての辺が ST パスの一部となっていることであると与えた．さらに二端子グラフが冗長であるとは次の条件と同義である．

[定理] 二端子グラフ G が冗長である ⇔ 二端子グラフ G に切断点[*2]C が存在し，その切断点 C で切断したグラフに S にも T にも連結していない頂点 P が存在する．

(証明) 必要条件は，S から P，P から T に向かうためにどちらも C を経由しなければならないため，P に接続している辺は S から T への単純パスの一部ではないことが分かり，二端子グラフは冗長である．一方で十分条件の場合はグラフが冗長であるならば，ST パスの一部に含まれない辺が存在し，ST パスの一部に含まれない辺の集合と補集合の双方の集合の辺が接続している頂点が存在する．もし、その頂点が切断点でなかったと仮定すると、その頂点から ST パスの一部になっていない辺を辿っていくことが出来る。それを続けると先ほどの切断点とは異なる頂点でかつ ST パスになっていない辺と ST パスになっている辺の双方が接続している頂点に到達する。これでは、ST パスになっていない辺も ST パスになってしまうため、背理法より仮定が誤りであり、その頂点は切断点となっており，切断した際に S にも T にも接続していない頂点が出現する．(証明終わり)

このことから，切断点を発見するアルゴリズム (Tarjan's algorithm[8]) を応用し，O(V+E) で二端子グラフが非冗長かどうかを判定することができる．

3.3.3　同型性判定

二端子グラフの同型性においては，2 つの同型性が存在する．S と T の対称性による同型性と，両端子以外の頂点番号の置換による同型性である．実装においては S,T の次数とその他の頂点の次数をそれぞれ次数列で管理し，次数列の組が同一のものについて，同一次数を持つ頂点の置換を行い，辺の接続が同じかどうかを判定し同型性判定を行っている．

3.4　多重辺の生成

まず，多重辺を生成する目的について述べる．今回，生成した二端子グラフの辺ごとにひとつのリテラルを割り当てるようプログラムを行なった．それは，複数の条件を一つの辺に割り当てると，割り当ての候補が増加し，必要な計算量が大きくなるためである．そのため，ある頂点間にトランジスタを並列に接続するパストランジスタ回路に相当するような多重辺を持つグラフを生成する必要性が生じた．また，多重辺は変数の数に応じて生成する必要のある本数が変化するため，直接多重辺を持つ二端子グラフを生成するのではなく，一度単純な二端子グラフを生成してから多重辺を付与している．変数の数より多くの多重辺を持つ場合，恒真条件，あるいは重複した条件となってしまうため，変数の個数に応じて生成される最大の多重辺は異なる．3 入力関数の場合は，3 並列より多い多重辺を生成する必要はない．

[*2] 削除することでグラフが非連結になるような頂点

ⓒ 2020 Information Processing Society of Japan

3.4.1 多重辺の生成と同型性

多重辺の生成と変数の関係については先の節で述べた．その際に，単純グラフに対して多重辺を付与すると，同型なグラフを生じる可能性がある．しかし，異なる単純グラフに多重辺を付与しても，同型になることはない．そのため，元の単純グラフが同一のもののみに対して同型性判定を行うことで，効率的に非同型な多重辺を持つ二端子グラフを生成することができる．

3.5 ラベルの割り当て

変数ラベルを割り当てる．先の節 3.4 でも述べたが，一つの辺に一つのリテラルを割り当てる．ここでは変数とは 3 入力関数のように入力として使用される文字種を示しており，リテラルはその変数の肯定リテラルあるいは否定リテラルを意味している．すなわち，辺に割り当てられるのはある変数の肯定リテラルまたは否定リテラルのみであり，複数の変数の和や積を辺に割り当てることは行わない．愚直に実装を行う場合，変数の個数 n, 辺の数 $|E|$ として $(2n)^{|E|}$ 通りの割り当てが存在するが，次節で説明する同値類の圧縮を用い計算を早めることができる．

3.5.1 同値類による高速化

同値類とは，論理関数において特定の操作をすることで変換することができる論理関数の集合である．今回用いる NP 同値類においては入力変数の交換と入力変数の否定によって同一とみなせる論理関数を同一の集合とみなしている．

同値類と二端子グラフの関係性は次のようなものである．二端子グラフに対してある変数ラベルを割り当て，ある論理関数が得られたとする．その論理関数の同値類は論理関数上で入力変数の入れ替えや入力変数の否定を取った操作と同様の操作を二端子グラフに対しても行うことで同値類を表す変数ラベル割り当てを得ることができる．具体例を図 6 に示す．図中左のグラフが $xy + \overline{x}z$ を表すことがわかったとする．論理関数上で同値類を得る操作とグラフ上で同値類を得る操作は同様である事が分かり，$z\overline{y} + \overline{z}x$ を表現するグラフもその論理関数上の操作が分かっていれば簡単に実現することができる．そのため同値類の対応関係を論理関数上で把握しておくことで，入れ替えを前提として，辺に付与するリテラルを全て試さずとも，同値類の割り当てを明らかにすることができる．このことにより，最大で $n! \cdot 2^n$ 通りの割り当てを一つの変数ラベルの割り当てで処理することができる．

3.6 表現論理関数の確認

割り当てられた変数ラベルを用いて，表現する論理関数の確認を行う．各変数の状態ごとに辺を有効化，あるいは無効化し，ST パスが存在するかどうかの探索を行う．愚直にそれぞれの入力変数ごとに ST パスの確認を行い，パストランジスタ回路の設計で問題となる設計時には意図していなかった長いパス（スネークパス）の導通による影響も踏まえた確認が行えている．

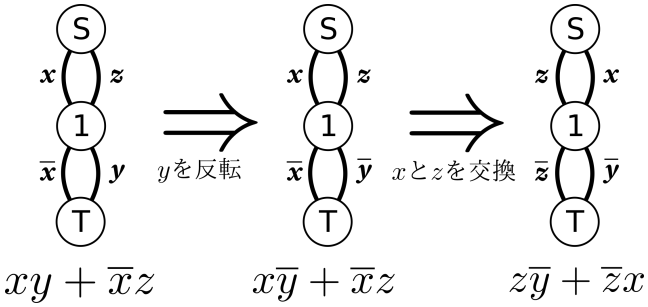


図 6 論理関数上の操作とグラフ上の操作の対応関係

4. 実験結果

4.1 実験手順と条件

3.1 節で述べた問題に対して実際にアルゴリズムを動作させることで検討を行った．本研究の手法で 8 辺までの非冗長非同型単純二端子グラフを列挙した．今回，3 入力関数を表すラベル付き二端子グラフを列挙するべく，さらに条件を満たす 8 辺までの多重辺を許す非同型非冗長二端子グラフの作成を行った．それらに対して，変数ラベルを割り当てそれぞれのグラフに対して表現する論理関数の確認を行った．3 入力論理関数の総数は 256 通りであるが，NP 同値類としては 22 種類である．

4.2 結果

まず，表 2 に，8 辺までの非同型非冗長な単純二端子グラフの生成を行なった結果を示す．x 印がついているところは，辺が少ないため全ての頂点を結ぶことができないためグラフが 0 個の箇所であり， – 印がついている箇所は頂点数に対して張ることのできる辺が不足しているためグラフが生成できないことを示している．これらに対して多重辺を許す非同型非冗長な二端子グラフの生成を行い，変数ラベルを割り当てることで探索を行ったところ，それぞれの同値類を表す最小の回路は図 9，および図 10 に示す結果となった．なお，複数の異なる回路が辺数最小となる場合は頂点数が最小となるものを選び, それでも同じ場合はそのうちの 1 つを表示している．また，図 7 はその素子数が最小素子数となっている同値類の個数をヒストグラムで表している．また，図 8 では，その素子数が最小素子数となっている 3 入力論理関数の個数をヒストグラムで表している．ただし，恒真および恒偽を表す関数については素子を使用しないため除いて表示している．今回すべての 3 入力論理関数は 8 素子以下で表現することが出来ることが確認された．また，8 素子を必要とする論理関数は同値類としては 1 種類，関数としては 2 種類であり，それは XOR および XNOR を表す回路であった．

ⓒ 2020 Information Processing Society of Japan

表 1 二端子グラフの辺数と変数ラベルを割り当てる所要時間

edge	5	6	7	8
time(s)	0.06	0.40	3.10	16.73

また一方で，今回それぞれの二端子グラフに変数ラベルを割り当て表現される論理関数が何通りあるかについても確認した．その結果，割り当てを変えることで全ての 3 入力論理関数を表現できる最小の二端子グラフは図 8 中のグラフ番号 15 に示す XOR を表現できるグラフと同一であることが明らかとなった．

4.3 実行時間

また，今回，アルゴリズムの実行においてボトルネックとなっている箇所が 2 箇所存在する．一つ目の箇所は，単純二端子グラフ生成の際の同型性判定である．これは，単純二端子グラフ生成時に同型なものを多く出力してしまうことに起因している．生成時にはシンプルな条件で枝狩りを行なっているが，さらに明らかに同型なものを枝狩りすることで，高速化を行うことができると考えられる．実際に問題となるのは，2 つ目の二端子グラフに対する変数ラベルの割り当ておよびその表現論理関数の確認である．NP 同値類による圧縮を説明したが，割り当て全体の計算量は変数の個数 n, 辺の数 $|E|$ として $(2n)^{|E|}$ であり，辺数に対して指数的に増加するのに対して，同値類による圧縮は最大 $n! \cdot 2^n$ であり，変数の階乗と 2 の変数乗の積で圧縮が行われる．そのため，1 本辺が増加した二端子グラフを取り扱う場合には，$2n$ 倍の計算時間を要することとなる．次の表 1 に辺数の異なる一つの二端子グラフに 3 変数のラベルを割り当て，表現する論理関数を確認した際に要した時間を示す．

5. まとめ

本研究では，パストランジスタ回路をモデル化した変数ラベル付き二端子グラフを段階を踏んで列挙を行うアルゴリズムを述べた．特に，二端子グラフの性質に着目し，二端子グラフの列挙をまず行い，得られたグラフに対して同値類の考え方を利用し高速化しながらラベル付き二端子グラフの列挙を行なった．そして，3 入力関数までの論理関数を表す最小の変数ラベル付き二端子グラフを明らかにした．今後の課題としては，4 入力関数を表現する最小のラベル付き二端子グラフ回路の列挙ならびに，4 入力関数を全て表現することのできる二端子グラフ回路の探索などが考えられる．今後は，さらなる高速化の検討などに努めることで，より大きなグラフに対して検討が行えるよう研究を行う．

謝辞 様々なご指導をくださいました川原純先生，岩政勇仁先生，そして京都大学 大学院情報学研究科 通信情報システム専攻 湊研究室の皆様方に感謝を申し上げます．なお本研究の一部は JSPS 科研費基盤 (A)（課題番号 20H00605），および JST CREST「学習/数理モデルに基づく時空間展開型アーキテクチャの創出と応用」の助成に基づく．

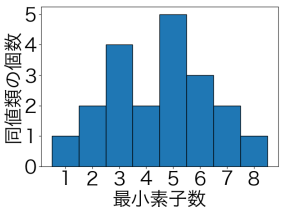


図 7 同値類を表す最小素子数とグラフ総数

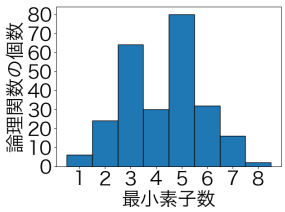


図 8 論理関数を表す最小素子数とグラフ総数

参考文献

[1] Buch, Narayan, Newton and Sangiovanni-Vincentelli: Logic synthesis for large pass transistor circuits, *1997 Proceedings of IEEE International Conference on Computer Aided Design(ICCAD)*, pp. 663–670 (1997).

[2] Yano, K., Sasaki, Y., Rikino, K. and Seki, K.: Top-down pass-transistor logic design, *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 6, pp. 792–803 (1996).

[3] Zimmermann, R. and Fichtner, W.: Low-power logic styles: CMOS versus pass-transistor logic, *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 7, pp. 1079–1090 (1997).

[4] Akers, S.: Binary Decision Diagrams, *IEEE Transactions on Computers*, Vol. 27, No. 06, pp. 509–516 (1978).

[5] Hellerman, L.: A Catalog of Three-Variable Or-Invert and And-Invert Logical Circuits, *IEEE Transactions on Electronic Computers*, Vol. EC-12, No. 3, pp. 198–223 (1963).

[6] McKay, B. D.: Isomorph-Free Exhaustive Generation, *Journal of Algorithms*, Vol. 26, No. 2, pp. 306 – 324 (1998).

[7] Denzumi, S., Horiyama, T., Kurita, K., Nakahata, Y., Suzuki, H., Wasa, K. and Yamazaki, K.: Enumeration and Random Sampling of Nonisomorphic Two-Terminal Series-Parallel Graphs, *IEICE Tech. Rep.*, COMP2018-17, Vol. 118, No. 216, pp. 55–62 (2018).

[8] Tarjan, R.: Depth-First Search and Linear Graph Algorithms, *SIAM Journal on Computing*, Vol. 1, No. 2, pp. 146–160 (1972).

表 2 非同型非冗長な単純二端子グラフの個数
x は辺不足, -は頂点不足による 0 を表す

		vertex								
		2	3	4	5	6	7	8	9	sum
edge	1	1	x	x	x	x	x	x	x	1
	2	-	1	x	x	x	x	x	x	1
	3	-	1	1	x	x	x	x	x	2
	4	-	-	3	1	x	x	x	x	4
	5	-	-	3	6	1	x	x	x	10
	6	-	-	1	14	11	1	x	x	27
	7	-	-	-	15	47	17	1	x	80
	8	-	-	-	8	100	123	26	1	258

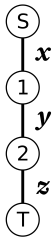
ⓒ 2020 Information Processing Society of Japan

グラフ番号：1

素子数	0	同値類	1
論理関数例	0(恒偽)		

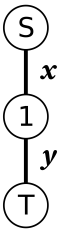
グラフ番号：2

素子数	3	同値類	8
論理関数例	xyz		



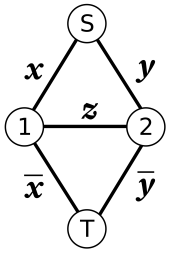
グラフ番号：3

素子数	2	同値類	12
論理関数例	xy		



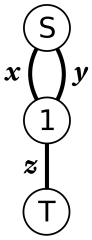
グラフ番号：4

素子数	5	同値類	12
論理関数例	$x\overline{y}z + \overline{x}yz$		



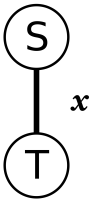
グラフ番号：5

素子数	3	同値類	24
論理関数例	$(x+y)z$		



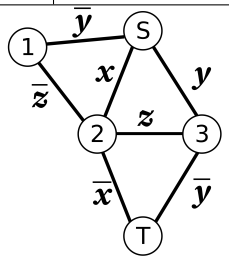
グラフ番号：6

素子数	1	同値類	6
論理関数例	x		



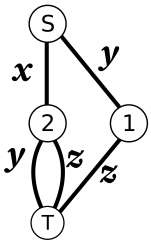
グラフ番号：7

素子数	7	同値類	8
論理関数例	$x\overline{y}z + \overline{x}yz + \overline{x}\,\overline{y}\,\overline{z}$		



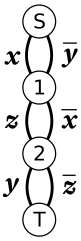
グラフ番号：8

素子数	5	同値類	8
論理関数例	$xy + yz + zx$		



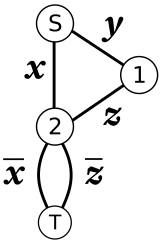
グラフ番号：9

素子数	6	同値類	4
論理関数例	$xyz + \overline{x}\,\overline{y}\,\overline{z}$		



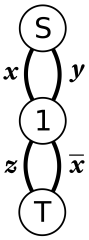
グラフ番号：10

素子数	5	同値類	24
論理関数例	$x\overline{z} + \overline{x}yz$		



グラフ番号：11

素子数	4	同値類	24
論理関数例	$xz + \overline{x}y$		



グラフ番号：12

素子数	3	同値類	24
論理関数例	$x + yz$		

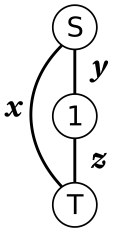
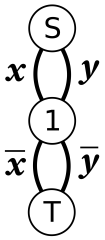


図 9　各 3 変数関数を表す変数ラベル付き二端子グラフ (1)

ⓒ 2020 Information Processing Society of Japan

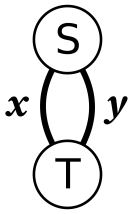
グラフ番号：13

素子数	4	同値類	6
論理関数例	$x\overline{y}+\overline{x}y$		



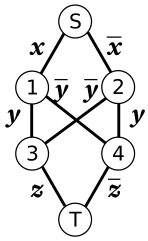
グラフ番号：14

素子数	2	同値類	12
論理関数例	$x+y$		



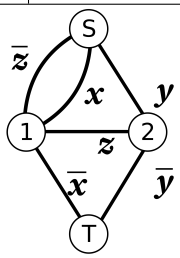
グラフ番号：15

素子数	8	同値類	2
論理関数例	$x\oplus y\oplus z$		



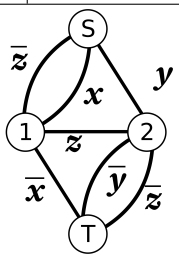
グラフ番号：16

素子数	6	同値類	24
論理関数例	$x\overline{y}z+\overline{x}(y+z)$		



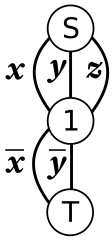
グラフ番号：17

素子数	7	同値類	8
論理関数例	$x\overline{y}z+\overline{x}(y+\overline{z})+y\overline{z}$		



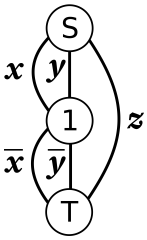
グラフ番号：18

素子数	5	同値類	24
論理関数例	$x\overline{y}+\overline{x}y+\overline{x}z$		



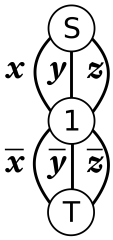
グラフ番号：19

素子数	5	同値類	12
論理関数例	$x\overline{y}+\overline{x}y+z$		



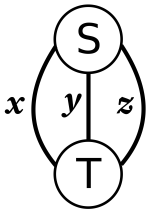
グラフ番号：20

素子数	6	同値類	4
論理関数例	$(x+y)\overline{z}+(\overline{x}+\overline{y})z$		



グラフ番号：21

素子数	3	同値類	8
論理関数例	$x+y+z$		



グラフ番号：22

素子数	0	同値類	1
論理関数例	1(恒真)		

図 10 各 3 変数関数を表す変数ラベル付き二端子グラフ (2)

ⓒ 2020 Information Processing Society of Japan