本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

論理設計
東野担当４回目
授業スライド
１０月２８日

OSAKA UNIVERSITY

基礎工学部情報科学科　東野輝夫

授業時間変更＆中間試験のお知らせ

- 長谷川先生のご都合で，長谷川先生担当の水曜３限の計算機言語の授業と水曜４限の東野の論理設計の授業を下記のように交換して実施します.
- １１月２５日（水）
  - ３限　計算機言語　→　論理設計　の授業に変更
  - ４限　論理設計　　（この日は２コマ続けて論理設計の授業を実施）
- １２月　９日（水）
  - ３限　計算機言語　（この日は２コマ続けて計算機言語の授業を実施）
  - ４限　論理設計　→　計算機言語　の授業に変更

- 東野担当の論理設計の中間試験について
  - １１月２５日（水）の４限（15:10 開始）に東野担当の論理設計の中間試験を実施します.
  - この日は３限に普通の授業を実施し，４限に試験を実施します．試験はCLE上で実施します．詳細は別途連絡しますが，下記で実施予定.
    - 15:10　試験の方法を説明（CLEビデオ or Zoom）
    - 15:20　CLEに試験問題を掲示し，制限時間を決めて時間内にレポート課題提出と同じ方法で答案をuploadしてもらうことで答案回収します.

授業計画の変更

長谷川先生との授業の入れ替えに伴い，章の説明の順番を変更します

- 授業計画：東野担当の授業計画を下記のように変更します．
  1. ドントケアを含む論理関数の簡単化（６章）
  2. フリップフロップとレジスタ（１０章）
  3. 同期式順序回路（Mealy型, Moore型順序回路）（１１章）
  4. カルノー図を用いた論理関数の簡単化（１章から５章の復習）
  5. 組合せ論理回路設計、よく用いられる組み合わせ回路（７章，８章）
  6. 加減算器とALU、順序回路の簡単化（９章，１２章前半）
  7. 演習
  8. 順序回路の簡単化、カウンタ（１２章後半，１３章）
  9. 中間試験（１章～１１章）
  10. ＩＣを用いた順序回路の実現（１５章）
  11. 演習
  12. ＣＰＵの設計（付録）
  13. ＣＰＵの設計，演習
  14. 乗算器と除算器（１４章）
  15. 期末試験（１２章～１５章，付録）

８コマ目の授業を11/25の３限に実施
９コマ目の試験を11/25の４限に実施
中間試験の範囲を１１章までに変更
期末試験の範囲を１２章以降に変更

質問について

- メールで随時問い合わせや質問にお答えしますので，何かあれば，higashino@ist.osaka-u.ac.jp までメールで質問して下さい.
- また、時間を決めてZoomなどを用いて質問にお答えすることも可能ですので、まずはメールで疑問点や問い合わせ事項などを連絡して下さい。

お願い

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

著作権保護

- この授業のテキスト（教科書）や授業スライド、授業ビデオの著作権保護に努めて下さい.
- この授業のビデオやスナップショットを録画したり，それらを他の人に転送したり，インターネット上で公開したりすることを禁止します.
- この授業で利用するスライドにはオーム社の教科書の図などが含まれているので，著作権保護の観点から，この授業スライドの公開につながる行為は謹んでください.
- 来年度は CLE を使ったメディア授業でなく，対面の授業ができることを期待していますが，今年度の演習課題の解答が事前に公開されたりすると，来年度の授業で同じ演習課題が使えなくなり，授業テキストの大幅な修正が必要になるため，協力をお願いします.

３回目の授業の
レポート課題の解答

３回目の授業のレポート課題 139頁の演習問題３

レポート課題

（課題３）

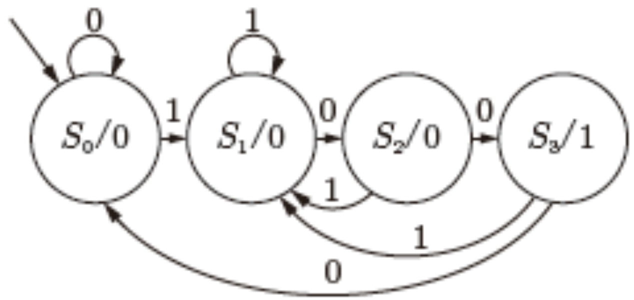
- パターン “100” を見つける文字列検出回路を Moore型順序回路 で設計せよ.
- 最終的な解答は教科書の末尾に書かれていますが，このレポート課題では状態遷移図をどう作り，その遷移図の S_0 に 00，S_1 に 01，S_2 に 11，S_3 に10 と割り当てると，状態遷移関数，出力関数がカルノー図でどう表されるかや，そのカルノー図から状態遷移関数，出力関数の最簡積和形をどう導出するかなどの計算過程を明記ください.
- 提出先：wordやpower pointなどで電子的に作成するか，紙に書いた解答をスマホで写真を取ったりスキャナーなどで読み取り、pdf や jpeg, gif などの形式で電子化して CLE にアップして下さい.
- 締切：10月27日(火) 23:59（次の授業の前日迄）.
- ４回目の授業は 10月28日(水) の15:10 から実施します．授業のビデオは当日の12:30以降に視聴できるようになります.

演習問題３解答（状態遷移図）

- パターン “100” を見つける文字列検出回路を Moore型順序回路 で設計せよ.

（解答）

- 1 が 1 回以上連続して入力された状態である S_1, 10 が連続して入力された状態である S_2, 100 が連続して入力された状態である S_3, そしてその他の状態 S_0 を定義し，これら 4 状態を使って，Moore型順序機械として実現すると，解図11･3 の状態遷移図を得る.
- S_0 に 00, S_1 に 01, S_2 に 11, S_3 に10 と割り当てると，状態遷移関数，出力関数は以下となる.



解図 11・3 “100” 文字列検出回路の状態遷移図

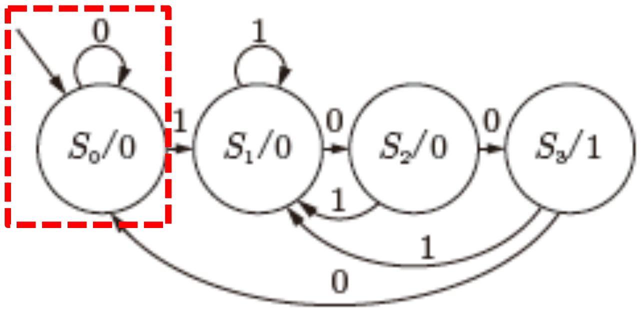
演習問題３解答（状態遷移図）

x＼q_1q_0

- パターン “100” を見つける文字列検出回路を Moore型順序回路 で設計せよ.

（解答）

- 1 が 1 回以上連続して入力された状態である S_1，10 が連続して入力された状態である S_2，100 が連続して入力された状態である S_3，そしてその他の状態 S_0 を定義し，これら 4 状態を使って，Moore 型順序機械として実現すると，解図11·3 の状態遷移図を得る.
- S_0 に 00，S_1 に 01，S_2 に 11，S_3 に10 と割り当てると，状態遷移関数，出力関数は以下となる.



解図 11・3　“100” 文字列検出回路の状態遷移図

q_1^+

x＼q_1q_0	00	01	11	10
0	0	1	1	0
1	0	0	0	0

q_0^+

x＼q_1q_0	00	01	11	10
0	0	1	0	0
1	1	1	1	1

z

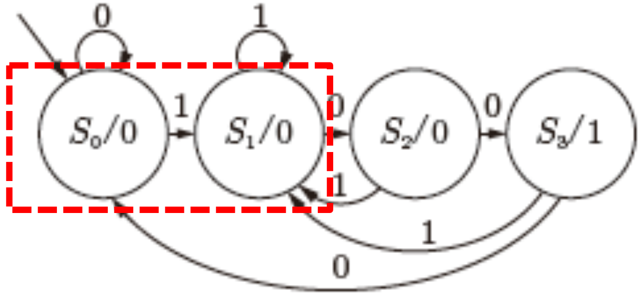
q_1＼q_0	0	1
0	0	0
1	1	0

演習問題３解答（状態遷移図）

- パターン “100” を見つける文字列検出回路を Moore型順序回路 で設計せよ.

（解答）

- 1 が 1 回以上連続して入力された状態である S_1，10 が連続して入力された状態である S_2，100 が連続して入力された状態である S_3，そしてその他の状態 S_0 を定義し，これら 4 状態を使って，Moore型順序機械として実現すると，解図11·3 の状態遷移図を得る.
- S_0 に 00，S_1 に 01，S_2 に 11，S_3 に10 と割り当てると，状態遷移関数，出力関数は以下となる.



解図 11・3 “100” 文字列検出回路の状態遷移図

q_1^+				
x＼q_1q_0	00	01	11	10
0	0	1	1	0
1	0	0	0	0

q_0^+				
x＼q_1q_0	00	01	11	10
0	0	1	0	0
1	1	1	1	1

z		
q_1＼q_0	0	1
0	0	0
1	1	0

演習問題３解答（状態遷移関数，出力関数）

- パターン “100” を見つける文字列検出回路を Moore型順序回路 で設計せよ.

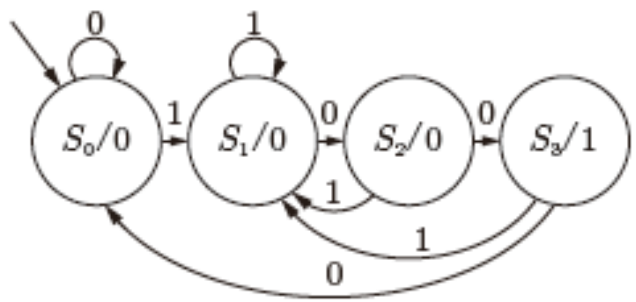
（解答）

- 状態遷移関数，出力関数は以下となる.

$$q_1^{+} = \bar{x} \cdot q_0$$

$$q_0^{+} = x \vee \overline{q_1} \cdot q_0$$

$$z = q_1 \cdot \overline{q_0}$$



解図 11・3 “100” 文字列検出回路の状態遷移図

q_1^{+}				
x＼q_1q_0	00	01	11	10
0	0	1	1	0
1	0	0	0	0

q_0^{+}				
x＼q_1q_0	00	01	11	10
0	0	1	0	0
1	1	1	1	1

z		
q_1＼q_0	0	1
0	0	0
1	1	0

カルノー図を用いた
論理関数の簡単化
（１章から５章の復習）

カルノー図

- 人間が最簡積和形を求める場合，2 章で学んだ カルノー図（Karnaugh map）を用いる方法が直感的に理解しやすい．n 変数論理関数のカルノー図の例を 図5·1 に示す．2 変数の場合2×2 のマス目，3 変数の場合 2×4 もしくは 4×2 のマス目，4変数の場合 4×4 のマス目，5 変数の場合 4×8 もしくは 8×4 のマス目となる．それぞれマス目の数は変数の数 n に対して 2^n となり，真理値表の行数に等しい．

x_1 \ x_0	0	1
0	(0)	(1)
1	(2)	(3)

（a） 2 変数

x_2 \ $x_1 x_0$	00	01	11	10
0	(0)	(1)	(3)	(2)
1	(4)	(5)	(7)	(6)

（b） 3 変数

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	(0)	(1)	(3)	(2)
01	(4)	(5)	(7)	(6)
11	(12)	(13)	(15)	(14)
10	(8)	(9)	(11)	(10)

（c） 4 変数

$x_4 x_3$ \ $x_2 x_1 x_0$	000	001	011	010	110	111	101	100
00	(0)	(1)	(3)	(2)	(6)	(7)	(5)	(4)
01	(8)	(9)	(11)	(10)	(14)	(15)	(13)	(12)
11	(24)	(25)	(27)	(26)	(30)	(31)	(29)	(28)
10	(16)	(17)	(19)	(18)	(22)	(23)	(21)	(20)

（d） 5 変数

図 5 · 1 2 変数から 5 変数のカルノー図

カルノー図

- 各マス目は真理値表の 1 行（つまり最小項）に対応しており，関数値が 0 となるか 1 となるかをカルノー図でも表現する.
- 例えば，図5·1 (a) の (0) は $(x_1,x_0) = (0,0)$，(b) の (2) は $(x_2,x_1,x_0) = (0,1,0)$，(c) の (13) は$(x_3,x_2,x_1,x_0) = (1,1,0,1)$，(d) の (28) のマス目は $(x_4,x_3,x_2,x_1,x_0) = (1,1,1,0,0)$ の最小項に対応する.
- ここで横あるいは縦に 2 変数が割り当てられている場合，00, 01, 11, 10 の順に添字が並んでおり，<u>隣どうしの添字は一変数の値のみが変化するように記載されている.</u>

x_1 \ x_0	0	1
0	(0)	(1)
1	(2)	(3)

（a） 2 変数

x_2 \ x_1x_0	00	01	11	10
0	(0)	(1)	(3)	(2)
1	(4)	(5)	(7)	(6)

（b） 3 変数

x_3x_2 \ x_1x_0	00	01	11	10
00	(0)	(1)	(3)	(2)
01	(4)	(5)	(7)	(6)
11	(12)	(13)	(15)	(14)
10	(8)	(9)	(11)	(10)

（c） 4 変数

x_4x_3 \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
00	(0)	(1)	(3)	(2)	(6)	(7)	(5)	(4)
01	(8)	(9)	(11)	(10)	(14)	(15)	(13)	(12)
11	(24)	(25)	(27)	(26)	(30)	(31)	(29)	(28)
10	(16)	(17)	(19)	(18)	(22)	(23)	(21)	(20)

（d） 5 変数

図 5 · 1 2 変数から 5 変数のカルノー図

カルノー図

- 隣接するマス目に注目する．例として 図5·1（c）のマス目 (13) (15) を考えると，(x_3,x_2,x_1,x_0) = (1,1,0,1), (1,1,1,1) に相当しており，一変数 (x_1) のみ 0, 1 が異なっている．0, 1 が異なっている変数の数をハミング距離と呼ぶ．マス目 (13) (15) ではハミング距離は 1 になる．
- また，カルノー図の左端と右端は隣接しており，ハミング距離= 1 の条件を満たしている（例えば (12) (14) のマス目，(x_3,x_2,x_1,x_0) = (1,1,0,0), (1,1,1,0) に相当）．同様にカルノー図の上端と下端も隣接している（例えば (1) (9) のマス目）．
- 図5·1(d) のように横に 3 変数が割り当てられている場合， x_1,x_0 の添字を左右対称となるように割り当てることで，隣接するマス目のハミング距離を 1 としている．これらは次節で説明する積項がループで表現できるために必要な条件である．

x_1 \ x_0	0	1
0	(0)	(1)
1	(2)	(3)

（a） 2 変数

x_2 \ x_1x_0	00	01	11	10
0	(0)	(1)	(3)	(2)
1	(4)	(5)	(7)	(6)

（b） 3 変数

x_3x_2 \ x_1x_0	00	01	11	10
00	(0)	(1)	(3)	(2)
01	(4)	(5)	(7)	(6)
11	(12)	(13)	(15)	(14)
10	(8)	(9)	(11)	(10)

（c） 4 変数

x_4x_3 \ $x_2x_1x_0$	000	001	011	010	110	111	101	100
00	(0)	(1)	(3)	(2)	(6)	(7)	(5)	(4)
01	(8)	(9)	(11)	(10)	(14)	(15)	(13)	(12)
11	(24)	(25)	(27)	(26)	(30)	(31)	(29)	(28)
10	(16)	(17)	(19)	(18)	(22)	(23)	(21)	(20)

（d） 5 変数

図 5・1 2 変数から 5 変数のカルノー図

カルノー図を用いた簡単化

カルノー図を用いた簡単化では以下の二つのポイントが重要である

- ポイント1：ループ（0 のマス目を内部に含まない 1 のみのマス目の長方形や正方形）が一つの積項に対応するため，ループの数が少ない方がよい．
- ポイント2：大きなループはリテラル数が少ない積項に対応するため，できるだけ大きなループがよい．

- 図5·2 (a) の例では，x_1x_0 に相当するループが x_1 に相当するループに内包されており，冗長になっている．この冗長な x_1x_0 のループを削除した結果が右に示されている．ループの数が積項数に相当するため，冗長なループの削除によって，積項数が 2 から 1 に削減されている（ポイント1）．

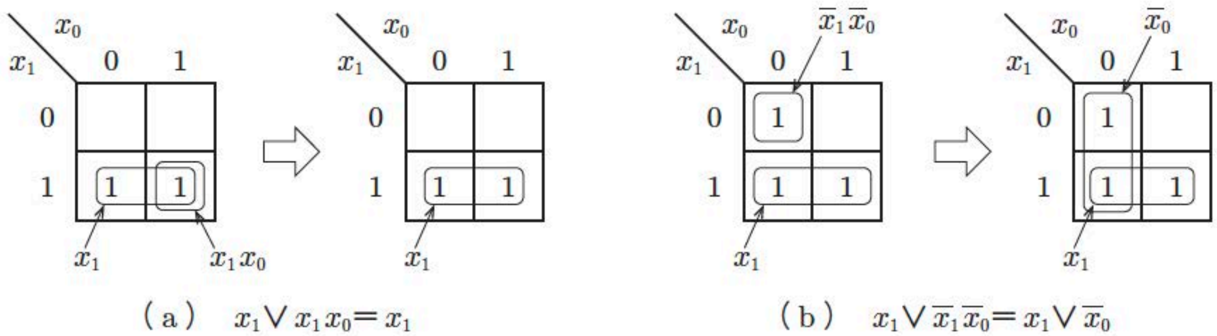


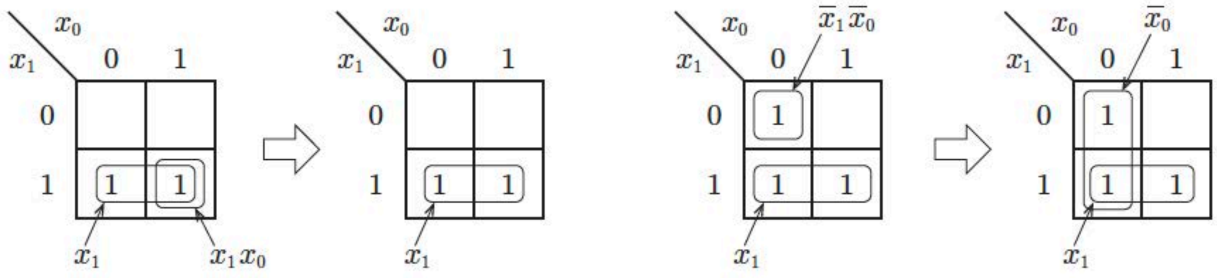
図 5・2 カルノー図の基本的な考え方

カルノー図を用いた簡単化

カルノー図を用いた簡単化では以下の二つのポイントが重要である

- ポイント1：ループ（0 のマス目を内部に含まない 1 のみのマス目の長方形や正方形）が一つの積項に対応するため，ループの数が少ない方がよい．
- ポイント2：大きなループはリテラル数が少ない積項に対応するため，できるだけ大きなループがよい．

- 図5·2 (b) の例では，$\bar{x}_1, \bar{x}_0$ に相当するループが $\bar{x}_0$ に相当するループに拡張されている．ループの拡張によってリテラル数の少ない積項に置き換わり，積項数は 2 と変わらないものの，リテラル数が 3 から 2 に削減されている（ポイント2）．



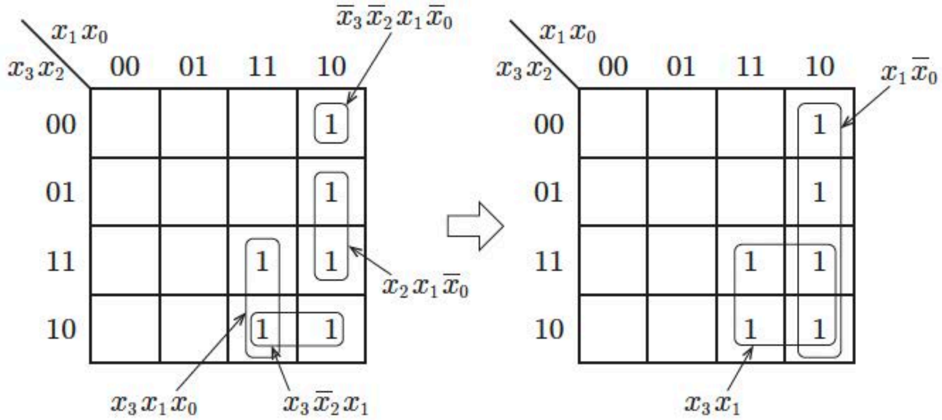
（a）　$x_1 \vee x_1 x_0 = x_1$　　（b）　$x_1 \vee \bar{x}_1 \bar{x}_0 = x_1 \vee \bar{x}_0$

図 5 · 2　カルノー図の基本的な考え方

カルノー図を用いた簡単化

<u>カルノー図を用いた簡単化では以下の二つのポイントが重要である</u>

- <u>ポイント1</u>：ループが一つの積項に対応するため，ループの数が少ない方がよい．ループの内部は 1 のマス目のみである．
- <u>ポイント2</u>：大きなループはリテラル数が少ない積項に対応するため，できるだけ大きなループがよい．
- 図5·2（c）の例では，ループの併合とループの拡張が組み合わされることで，すべての最小項を二つの大きなループで過不足なく覆っている．ループの数が 4 から 2 に，つまり積項数が 4 から 2 に減少している（ポイント1）．リテラル数も 13 から 4 に減少している（ポイント2）．



（c）　$x_3x_1x_0 \vee x_3\bar{x}_2x_1 \vee x_2x_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2x_1\bar{x}_0 = x_3x_1 \vee x_1\bar{x}_0$

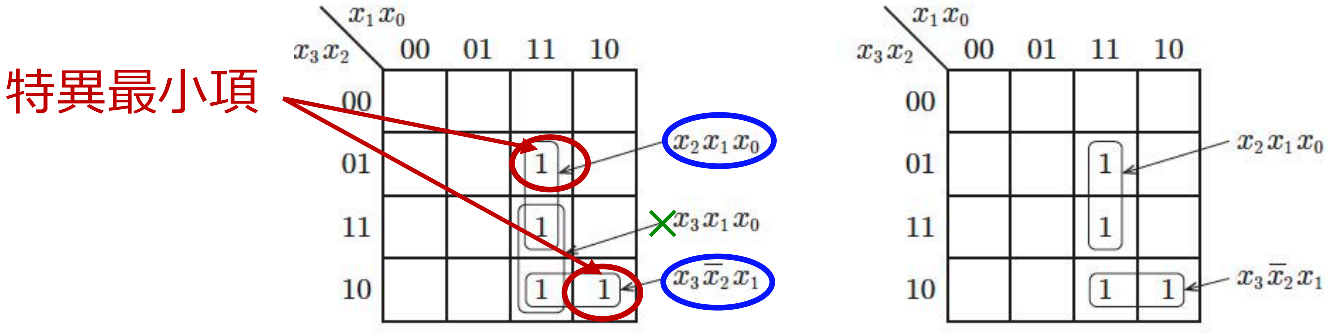
図 5 · 2　カルノー図の基本的な考え方

簡単化方法

- カルノー図を用いて論理関数 f の最簡積和形を求める場合，次の二つのステップが必要である.
  - ステップ1：f のすべての主項を求める
  - ステップ2：f の主項による最小被覆を求める
- 主項（prime implicant）とは，f に包含される積項のうち他の積項に包含されない積項である．直感的には f からはみ出さないできるだけ大きなループである（縦横のセルの数がそれぞれ 2 のべき乗である必要がある）．
- 包含関係：論理関数 f と積項 p について，p に含まれるすべての最小項（カルノー図中の 1）に対して f が 1 となるとき，f は p を包含しているという．また積項 p_1, p_2 があり，p_2 に含まれるすべての最小項が p_1 にも含まれているとき，p_1 は p_2 を包含しているという.
- ステップ2 では f のすべての最小項を最小数の主項で覆った最小被覆を求める.このとき一つの主項でしか被覆されていない最小項（特異最小項）に注目すると最小被覆が考えやすい（詳細は次章で説明する）．

例題5・2

- 図5·3 のカルノー図で表された論理関数 f の最簡積和形を求めよ.



（a） ステップ1　（b） ステップ2

図 5・4 簡単化のステップ

(解答)

- まずステップ 1 では f のすべての主項を求める．図5·4 (a) に求めた主項を示す．三つの主項 $x_2x_1x_0$, $x_3x_1x_0$, $x_3\bar{x}_2x_1$ が見つかっている．
- ステップ2 では主項による最小被覆を求める．図5·4 (b) に求めた最小被覆を示す．この例では，特異最小項 <u>$((x_3,x_2,x_1,x_0) = (0,1,1,1), (1,0,1,0))$</u> をそれぞれ含む二つの主項 $x_2x_1x_0$, $x_3\bar{x}_2x_1$ で f に含まれているすべての最小項が被覆できており，主項 $x_3x_1x_0$ は不要である．これにより最簡積和形は $f = x_2x_1x_0 \vee x_3\bar{x}_2x_1$ と求まる．
- ステップ1 は最簡積和形に含まれる可能性がある積項の列挙に相当し，ステップ2 は必要最小限の積項を選択する手続きである．

必須主項（あるいは必須項）

- ある最小項を被覆している主項が一つしかないとき，その主項を必須主項（あるいは必須項）（essential prime implicant）と呼び，その最小項を特異最小項と呼ぶ.

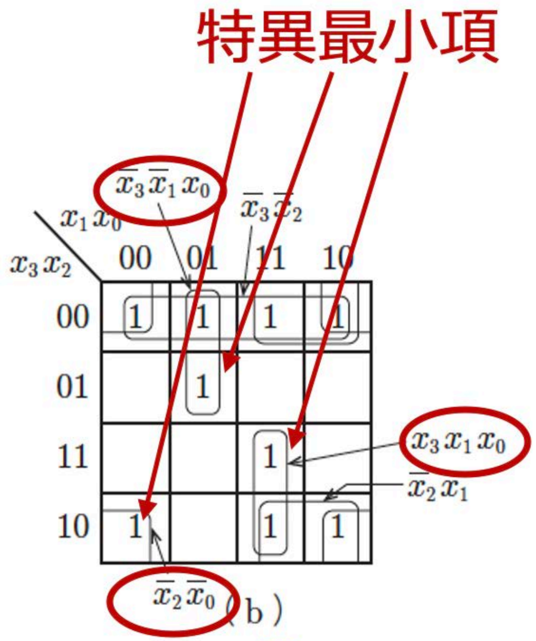


図 5・8 例題と解答

コンセンサスとリゾルベント

コンセンサス

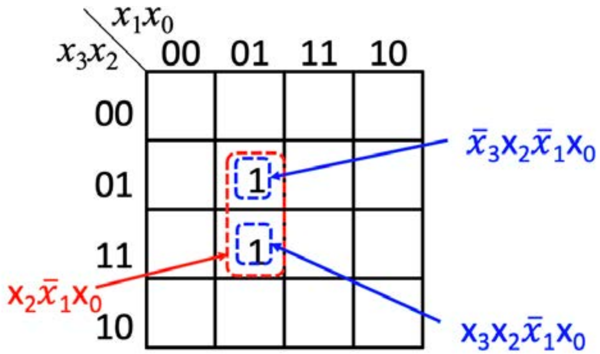
- 積項 t1 と t2 のコンセンサス(consensus) con(t1, t2) とは：
  - ただ一組の変数だけ肯定と否定が含まれているとき，それを除いて，t1 と t2 に含まれるリテラル (命題記号またはその否定) の ∧ をとったもの (重複するリテラルは一つにまとめる).

  (リテラルp と¬p は相補的であるともいう．)

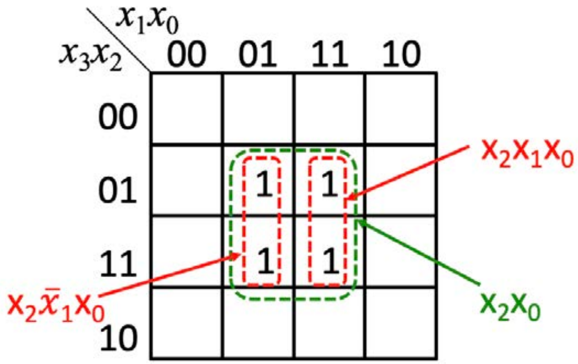
  - $\neg u \wedge w \wedge x \wedge \neg y$ と $\neg u \wedge w \wedge x \wedge y \wedge z$ のコンセンサスは $\neg u \wedge w \wedge x \wedge z$
  - $\neg u \wedge w \wedge x \wedge \neg y$ と $\neg u \wedge \neg w \wedge x \wedge y \wedge z$ のコンセンサスはない.
  - $\neg u \wedge w \wedge x \wedge \neg y$ と $\neg u \wedge w \wedge z$ のコンセンサスはない.

カルノー図の性質

- カルノー図上の隣り合うセルの論理式を考えてみよう

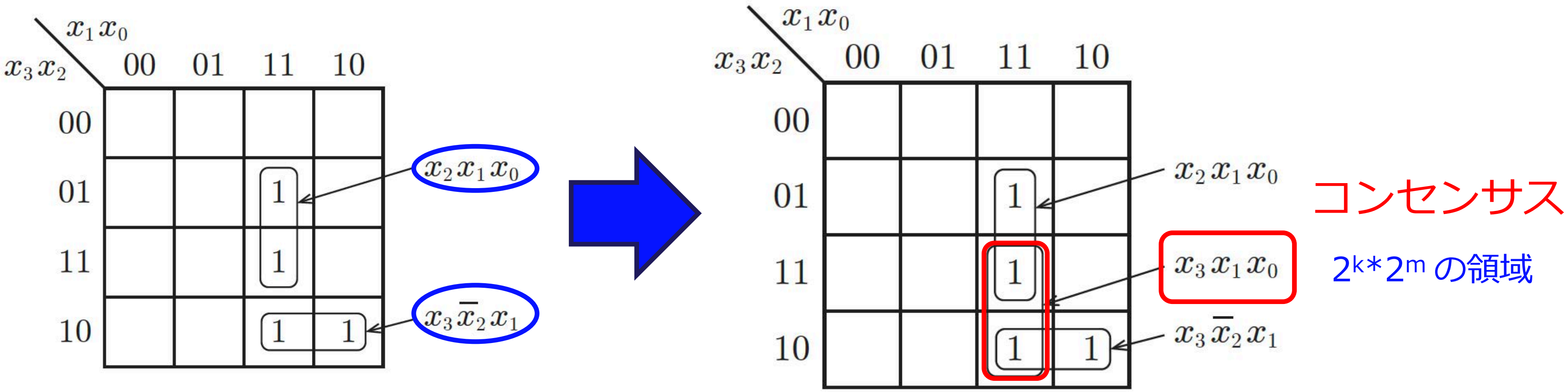


- 隣り合う２つのセルの論理式（論理積）を比べると一つの変数のみ肯定リテラルと否定リテラルになり、残りは同じリテラルである（ハミング距離＝１）
- ２つのセルを合わせたセルの論理式は肯定・否定リテラルを取り除いた論理式になる（２つの論理式のコンセンサスが得られる）
- ２つのセルを合わせたセル同士も同じ性質が成り立つ（より大きくても同じ）



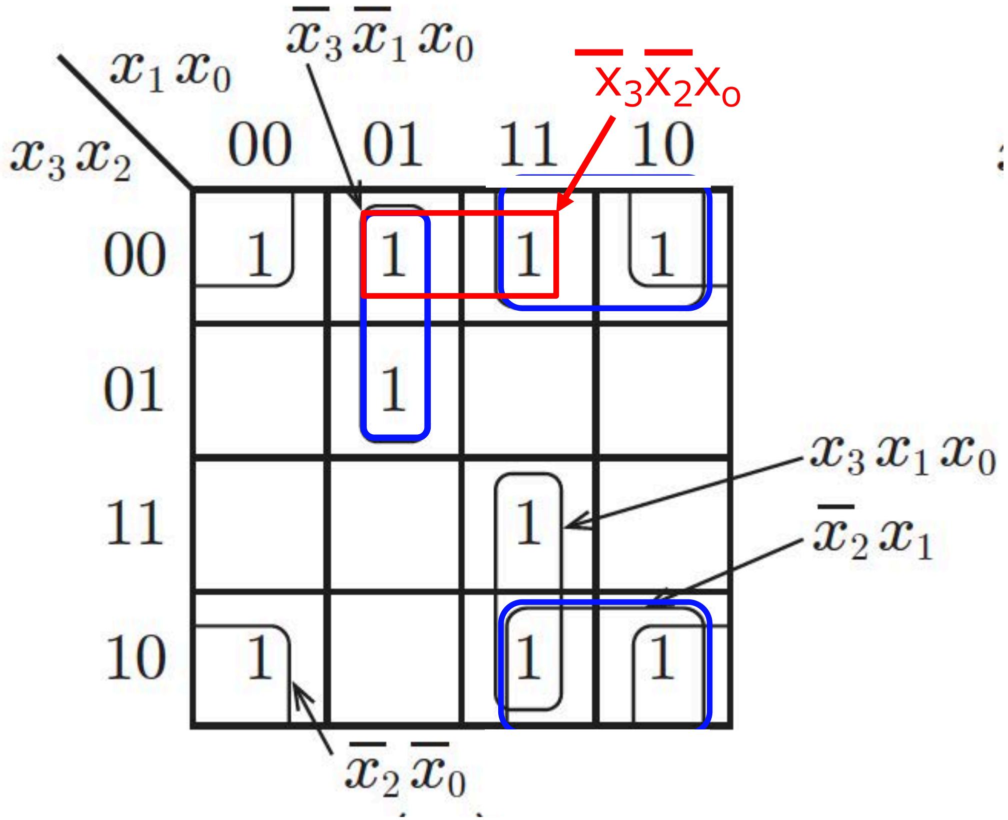
コンセンサスの性質

- カルノー図上で考えると， t1 と t2 のコンセンサスが存在する場合，t1 と t2 がおおう領域は必ず隣接している．また， t1 と t2 のコンセンサスがおおう領域は， t1 と t2 がおおう領域と交差し， t1 と t2 がおおう領域全体の中に含まれる（ con(t1,t2) ≤ t1 ∨ t2 ）．



コンセンサスの性質

- カルノー図上で考えると， t1 と t2 のコンセンサスが存在する場合，t1 と t2 がおおう領域は必ず隣接している．また， t1 と t2 のコンセンサスがおおう領域は， t1 と t2 がおおう領域と交差し， t1 と t2 がおおう領域全体の中に含まれる（ con(t1,t2) ≤ t1 ∨ t2 ）．



$$\mathrm{con}(\bar{x}_3 x_1 \bar{x}_0, x_2 x_1)$$

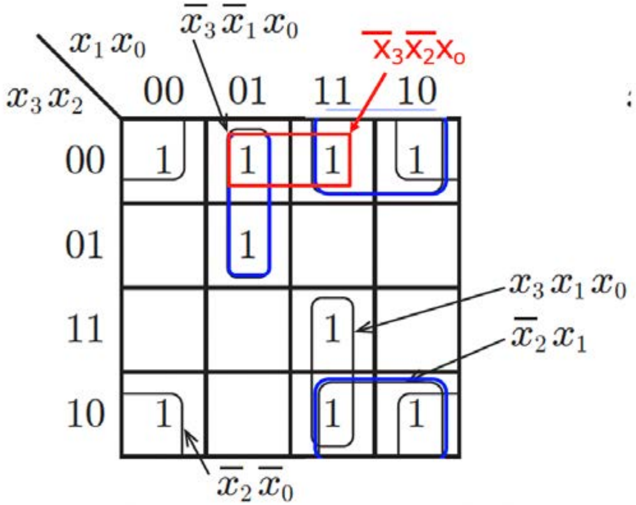
$$= \bar{x}_3 x_2 \bar{x}_0 \le \bar{x}_3 x_1 \bar{x}_0 \vee x_2 x_1$$

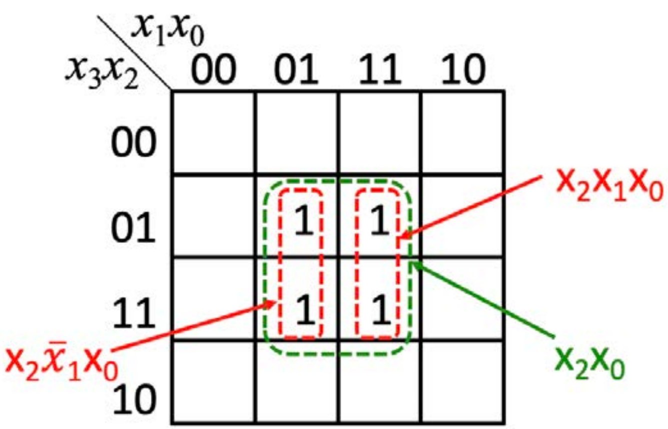
$2^k * 2^m$ の領域

OSAKA UNIVERSITY

コンセンサスの性質

- カルノー図上で t1 と t2 がおおう領域が隣接している場合，t1 と t2 を跨ぐ領域で 2^k*2^m の領域で最も大きな領域が t1 と t2 のコンセンサス con(t1,t2) として生成できる（ con(t1,t2) ≤ t1 ∨ t2 ）．
- t1 と t2 がおおう領域サイズが同じで，それらが一つの同じ長さの辺で隣接している場合，con(t1,t2) がおおう領域は t1 ∨ t2 がおおう領域と同じになる（ con(t1,t2) ＝ t1 ∨ t2 ）．
- f ＝ t1 ∨ t2 ∨ … ∨ tn とすると（各 tj は主項でなくてもよい），それらのコンセンサスをすべて生成すると， f のすべての主項が生成できる．また，それらのコンセンサスをすべて f に論理和した論理式を f′＝ f ∨ Σ(con(ti,tj)) とすると，con(t1,t2) ≤ t1 ∨ t2 より，f ＝ f′ が成り立つ．





OSAKA UNIVERSITY

追加資料

コンセンサスで主項を生成

- コンセンサスを作ることで，主項の集合を生成することができる．
  - 図(a)の３つの積項の論理和 $f = x_2\bar{x}_1 \vee x_3x_2x_0 \vee \bar{x}_3x_2x_1$ について，図(b)の $\mathrm{con}(x_3x_2x_0, \bar{x}_3x_2x_1) = x_2x_1x_0$ を生成

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	0
10	0	0	0	0

図(a) $f = x_2\bar{x}_1 \vee x_3x_2x_0 \vee \bar{x}_3x_2x_1$

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	0
10	0	0	0	0

図(b) $\mathrm{con}(x_3x_2x_0, \bar{x}_3x_2x_1) = x_2x_1x_0$

コンセンサスで主項を生成

- コンセンサスを作ることで，主項の集合を生成することができる．
  - 図(c)の con($x_2\bar{x}_1$, $x_2x_1x_0$) = x_2x_0 を生成
  - 包含関係を使って，包含される積項を取り除くと，図(d)の
    f = $x_2\bar{x}_1 \lor x_2x_0 \lor \bar{x}_3x_2x_1$ が生成される（主項の集合が得られる）

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	0
10	0	0	0	0

図(c) con($x_2\bar{x}_1$,$x_2x_1x_0$) = x_2x_0

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	0
10	0	0	0	0

図(d) f = $x_2\bar{x}_1 \lor x_2x_0 \lor \bar{x}_3x_2x_1$

クワイン・マクラスキー法

キューブ表現

- 主項の生成に積項のキューブ表現を用いる．論理変数 x_i に対する係数a_i を考え，a_i =1, 0, - のときにそれぞれ x_i, $\bar{x}_i$, 1 を表現するものとする．積項に対する係数の列挙$a_1a_2,\cdots,a_n$ を積項のキューブ表現と呼ぶ（n は論理変数の数）．
- 例えば 4 変数の論理関数において積項 $x_3x_2\bar{x}_1x_0$ のキューブ表現は 1101，x_2x_1 のキューブ表現は -11- である．
- 二つの積項のキューブ表現において (n−1) か所が同じで 1 か所だけが 0/1 と異なっているとき，これらの積項はカルノー図上で隣接している．これらの隣接した積項の論理和のキューブ表現は，異なっていた箇所を - に置き換えたものになる．この操作をキューブの併合という．
- 例えば，1-10 と1-00 を併合すると，1--0 となる．これは

  $$x_3x_1\bar{x}_0 \vee x_3\bar{x}_1\bar{x}_0 = x_3\bar{x}_0$$

  に相当する．
- キューブの併合は同じサイズの積項同士のコンセンサスの生成と同じ

クワイン・マクラスキー法（例題6・3）

<u>すべての主項を求める手続き</u>

- <u>ステップ1</u>：すべての最小項に対してキューブ表現を求める．キューブに含まれる 1 の個数によって昇順にグループ分けし，第一段階のリストを生成する．
- <u>ステップ2</u>：リストの第一グループに含まれるキューブと第二グループに含まれるキューブを比較し 1 か所だけ異なる場合には，それらのキューブにチェックを付ける．それらを併合して得られるキューブを次の段階のリストの第一グループに入れる．
- 同様に第二グループと第三グループに対して比較し，併合結果は次の段階のリストの第二グループに入れる．この隣どうしのグループに対する比較・併合を最後のグループまで行う．
- <u>ステップ3</u>：次の段階のリストにグループが二つ以上あれば，ステップ2 に戻る．
- <u>ステップ4</u>：チェックのついていないキューブが主項である．

第一段階のリスト

	x_3	x_2	x_1	x_0	最小項	
第一グループ	0	0	0	0	(0)	v
第二グループ	0	0	1	0	(2)	v
	1	0	0	0	(8)	v
第三グループ	0	0	1	1	(3)	v
	0	1	0	1	(5)	v
	0	1	1	0	(6)	v
	1	0	0	1	(9)	v
第四グループ	0	1	1	1	(7)	v

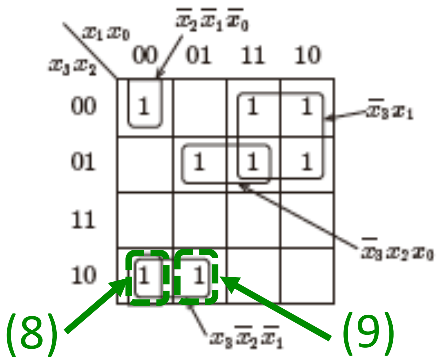
第二段階のリスト

x_3	x_2	x_1	x_0	最小項	
0	0	−	0	(0, 2)	
−	0	0	0	(0, 8)	
0	0	1	−	(2, 3)	v
0	−	1	0	(2, 6)	v
1	0	0	−	(8, 9)	
0	−	1	1	(3, 7)	v
0	1	−	1	(5, 7)	
0	1	1	−	(6, 7)	v

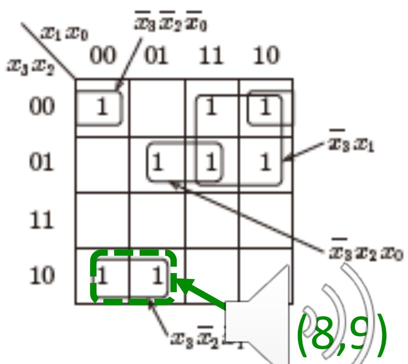
第三段階のリスト

x_3	x_2	x_1	x_0	最小項
0	−	1	−	(2, 3, 6, 7)

図 6・6　例題 6.3 の主項の生成



（a）f_a のカルノー図



（b）f_a のカルノー図

OSAKA UNIVERSITY

クワイン・マクラスキー法（例題6・3）

すべての主項を求める手続き

- ステップ1：すべての最小項に対してキューブ表現を求める．キューブに含まれる 1 の個数によって昇順にグループ分けし，第一段階のリストを生成する．
- ステップ2：リストの第一グループに含まれるキューブと第二グループに含まれるキューブを比較し 1 か所だけ異なる場合には，それらのキューブにチェックを付ける．それらを併合して得られるキューブを次の段階のリストの第一グループに入れる．
- 同様に第二グループと第三グループに対して比較し，併合結果は次の段階のリストの第二グループに入れる．この隣どうしのグループに対する比較・併合を最後のグループまで行う．
- ステップ3：次の段階のリストにグループが二つ以上あれば，ステップ2 に戻る．
- ステップ4：チェックのついていないキューブが主項である．

第一段階のリスト

	x_3	x_2	x_1	x_0	最小項	
第一グループ	0	0	0	0	(0)	v
第二グループ	0	0	1	0	(2)	v
	1	0	0	0	(8)	v
第三グループ	0	0	1	1	(3)	v
	0	1	0	1	(5)	v
	0	1	1	0	(6)	v
	1	0	0	1	(9)	v
第四グループ	0	1	1	1	(7)	v

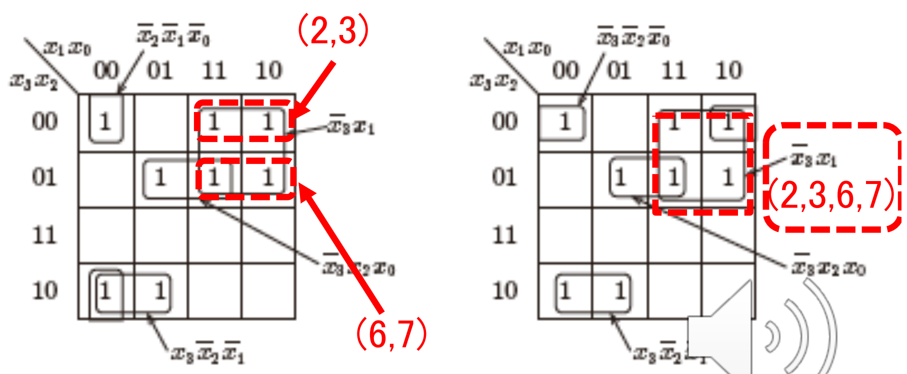
第二段階のリスト

x_3	x_2	x_1	x_0	最小項	
0	0	−	0	(0, 2)	
−	0	0	0	(0, 8)	
0	0	1	−	(2, 3)	v
0	−	1	0	(2, 6)	v
1	0	0	−	(8, 9)	
0	−	1	1	(3, 7)	v
0	1	−	1	(5, 7)	
0	1	1	−	(6, 7)	v

第三段階のリスト

x_3	x_2	x_1	x_0	最小項
0	−	1	−	(2, 3, 6, 7)

図 6・6　例題 6-3 の主項の生成



（a） f_a のカルノー図　（b） f_a のカルノー図

クワイン・マクラスキー法（例題6・3）

すべての主項を求める手続き

- ステップ1：すべての最小項に対してキューブ表現を求める．キューブに含まれる 1 の個数によって昇順にグループ分けし，第一段階のリストを生成する．
- ステップ2：リストの第一グループに含まれるキューブと第二グループに含まれるキューブを比較し 1 か所だけ異なる場合には，それらのキューブにチェックを付ける．それらを併合して得られるキューブを次の段階のリストの第一グループに入れる．
- 同様に第二グループと第三グループに対して比較し，併合結果は次の段階のリストの第二グループに入れる．この隣どうしのグループに対する比較・併合を最後のグループまで行う．
- ステップ3：次の段階のリストにグループが二つ以上あれば，ステップ2 に戻る．
- ステップ4：チェックのついていないキューブが主項である．

第一段階のリスト

	x_3	x_2	x_1	x_0	最小項	
第一グループ	0	0	0	0	(0)	v
第二グループ	0	0	1	0	(2)	v
	1	0	0	0	(8)	v
第三グループ	0	0	1	1	(3)	v
	0	1	0	1	(5)	v
	0	1	1	0	(6)	v
	1	0	0	1	(9)	v
第四グループ	0	1	1	1	(7)	v

第二段階のリスト

x_3	x_2	x_1	x_0	最小項	
0	0	–	0	(0, 2)	
–	0	0	0	(0, 8)	
0	0	1	–	(2, 3)	v
0	–	1	0	(2, 6)	v
1	0	0	–	(8, 9)	
0	–	1	1	(3, 7)	v
0	1	–	1	(5, 7)	
0	1	1	–	(6, 7)	v

第三段階のリスト

x_3	x_2	x_1	x_0	最小項
0	–	1	–	(2, 3, 6, 7)

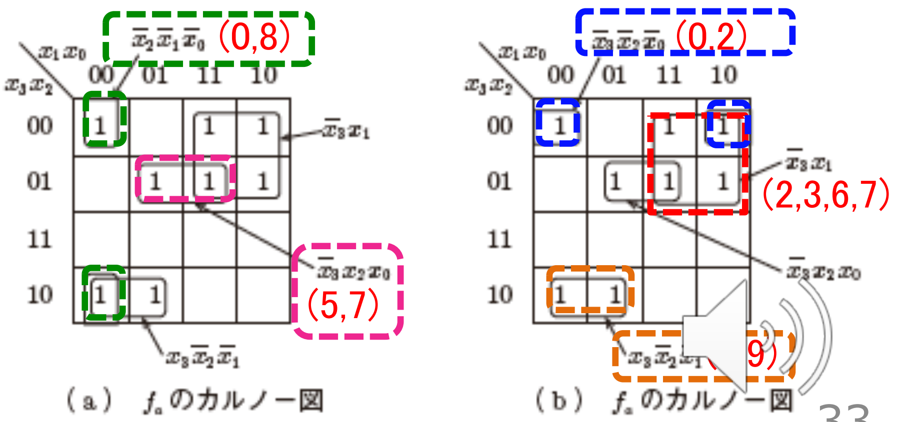


図 6・6 例題 6.3 の主項の生成

リゾルベント

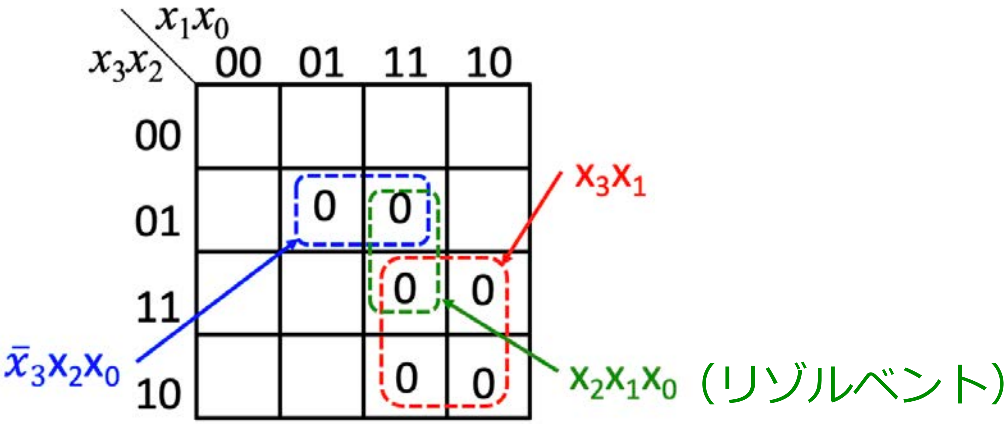
- 節(和項) t1 と t2 のリゾルベント(resolvent) res(ti, tj) とは：
  - ただ一組の変数だけ肯定と否定が含まれているとき，それを除いて，t1 と t2 に含まれるリテラル(命題記号またはその否定)の ∨ をとったもの(重複するリテラルは一つにまとめる).

  (リテラルp と¬p は相補的であるともいう．)

  - ¬u ∨ w ∨ x ∨ ¬y と ¬u ∨ w ∨ x ∨ y ∨ z のリゾルベントは ¬u ∨ w ∨ x ∨ z
  - ¬u ∨ w ∨ x ∨ ¬y と ¬u ∨ ¬w ∨ x ∨ y ∨ z のリゾルベントはない.
  - ¬u ∨ w ∨ x ∨ ¬y と ¬u ∨ w ∨ z のリゾルベントはない.

リゾルベントの性質

- カルノー図上で考えると， t1 と t2 のリゾルベントが存在する場合， t1 と t2 の 0 がおおう領域は必ず隣接している．また， t1 と t2 のリゾルベントの 0 がおおう領域は， t1 と t2 の 0 がおおう領域と交差し， t1 と t2 の 0 がおおう領域全体の中に含まれる（ t1 ∧ t2 ≤res(t1, t2) ）．
- t1 と t2 の 0 がおおう領域サイズが同じで，それらが一つの同じ長さの辺で隣接している場合，res(t1, t2) がおおう領域は t1 ∧ t2 がおおう領域と同じになる（ t1 ∧ t2 ≤ res(t1, t2) ）．
- f = t1 ∧ t2 ∧ … ∧ tn とすると（各 tj は主節でなくてもよい），それらのリゾルベントをすべて生成すると， f のすべての主節が生成できる．また，それらのリゾルベントをすべて f に論理積した論理式を f′= f ∧ Π(res(ti,tj)) とすると， t1 ∧ t2 ≤ res(t1, t2) より，f = f′ が成り立つ．



恒真性や恒偽性の判定

恒真性，充足可能性

（a） 恒真性

- n 変数論理関数 $f(x_0,\cdots, x_{n-1})$ に対して$\forall\ x_0,\cdots, x_{n-1}\ f(x_0,\cdots, x_{n-1})$ が真のとき，すなわち，V^n に属する任意の値 $(v_0,\cdots, v_{n-1})$ を変数 $x_0,\cdots, x_{n-1}$ に代入した式 $f(v_0,\cdots, v_{n-1})$ の値が常に真であるとき，論理関数 $f(x_0,\cdots, x_{n-1})$ は 恒真（valid）であるという.

（b） 充足可能性

- n 変数論理関数 $f(x_0,\cdots, x_{n-1})$ に対して$\exists x_0,\cdots, x_{n-1}\ f(x_0,\cdots, x_{n-1})$ が真のとき，すなわち，ある値 $(v_0,\cdots, v_{n-1}) \in V^n$ に対して，式 $f(v_0,\cdots, v_{n-1})$ の値が真であるとき，論理関数 $f(x_0,\cdots, x_{n-1})$ は充足可能（satisfiable）であるという.
- また，そのような値が存在しないとき，すなわち

$$\neg(\exists x_0,\cdots, x_{n-1}\ f(x_0,\cdots, x_{n-1}))$$

が真のとき $f(x_0,\cdots, x_{n-1})$ は充足不能（unsatisfiable）であるという.

論理式 P の恒真性の判定法

A) 「全ての解釈 (k 個の命題記号があれば 2^k 通り) に対して，それぞれの解釈のもとで P は真」なら恒真．そうでなければ (すなわち，P が偽になるような解釈があれば)，P は恒真ではない．全ての解釈についてそれぞれ真理値を計算することによって判定できる．

B) 等価な積和形に変換し，それに対しコンセンサス法を適用し，1 なる項が出れば P は恒真，そうでなければ (1 なる項が出なければ) P は恒真ではない．

C) 全ての解釈に対して ¬P が偽であるなら，P は恒真．そうでなければ，P は恒真ではない．全ての解釈について調べれば良い．

D) ¬P を等価な和積形に変換し，それに対しリゾルベント法を適用し，0 なる節(和項) が出れば， ¬P は恒偽，したがって P は恒真，そうでなければ (0 なる節が出なければ)， ¬P は恒偽でない，したがって P は恒真でない．

- (A) や(C) の方法は，命題記号の数が 50 とか 100 とかであれば，現実には実行不可能．

練習問題

問題

- コンセンサスを次々と求めていくことにより，次の論理関数 f が恒真であることを証明せよ.

$$f = (A \cdot B) \vee (A \cdot \overline{B}) \vee \overline{A}$$

練習問題（解答）

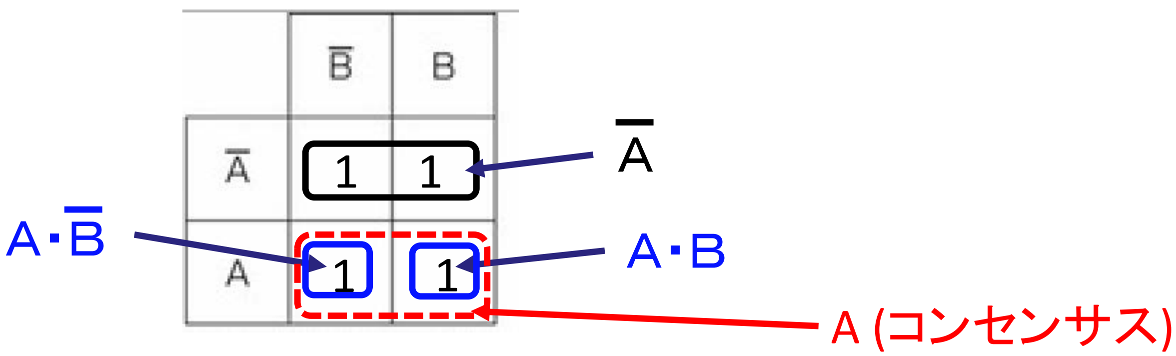
問題

- コンセンサスを次々と求めていくことにより，次の論理関数 f が恒真であることを証明せよ.

$$f = (A \cdot B) \vee (A \cdot \overline{B}) \vee \overline{A}$$
$$= (A \cdot B) \vee (A \cdot \overline{B}) \vee \overline{A} \vee A$$

$$\mathrm{con}(t1, t2) \leq t1 \vee t2$$

$$A = \mathrm{con}(A \cdot B, A \cdot \overline{B}) \leq A \cdot B \vee A \cdot \overline{B}$$



練習問題（解答）

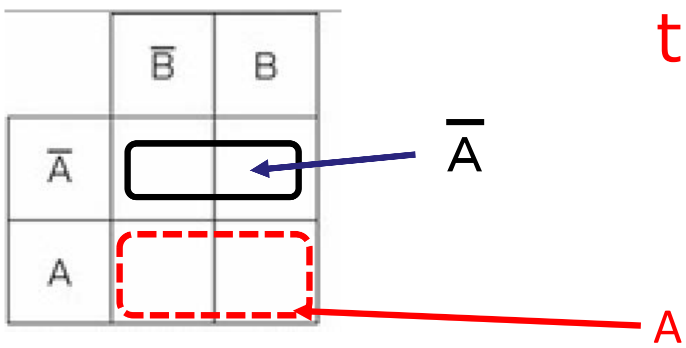
問題

- コンセンサスを次々と求めていくことにより，次の論理関数 f が恒真であることを証明せよ.

$$\begin{aligned} f &= (A \cdot B) \vee (A \cdot \overline{B}) \vee \overline{A} \\ &= (A \cdot B) \vee (A \cdot \overline{B}) \vee \overline{A} \vee A \\ &= (A \cdot B) \vee (A \cdot B) \vee \overline{A} \vee A \vee \text{true} \\ &= \text{true} \end{aligned}$$

$$\text{con}(t1, t2) \le t1 \vee t2$$

$$\text{true} = \text{con}(\overline{A}, A) \le \overline{A} \vee A$$



練習問題

問題

- リゾルベントを次々と求めていくことにより，次の論理関数 g が恒偽であることを証明せよ.

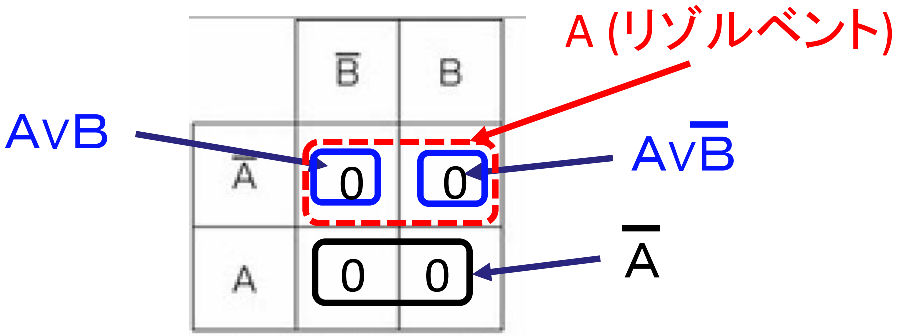
$$g = (A \lor B) \land (A \lor \overline{B}) \land \overline{A}$$

練習問題（解答）

問題

- リゾルベントを次々と求めていくことにより，次の論理関数 g が恒偽であることを証明せよ.

$$g = (A \lor B) \land (A \lor \overline{B}) \land \overline{A}$$
$$= (A \lor B) \land (A \lor \overline{B}) \land \overline{A} \land A$$



$$t1 \land t2 \leq res(t1, t2)$$

$$(A \lor B) \land (A \lor \overline{B})$$
$$\leq res((A \lor B), (A \lor \overline{B}))$$
$$= A$$

練習問題（解答）

問題

- リゾルベントを次々と求めていくことにより，次の論理関数 g が恒偽であることを証明せよ.

$$
\begin{aligned}
g &= (A \lor B) \land (A \lor \overline{B}) \land \overline{A} \\
&= (A \lor B) \land (A \lor \overline{B}) \land \overline{A} \land A \\
&= (A \lor B) \land (A \lor B) \land \overline{A} \land A \land \text{false} \\
&= \text{false}
\end{aligned}
$$

	$\overline{B}$	B
$\overline{A}$	0	0
A	0	0

A

$\overline{A}$

$t1 \land t2 \leq res(t1, t2)$

$\overline{A} \land A \leq res(\overline{A}, A)$
$= \text{false}$

導出原理

- 導出節(リゾルベント)を作っていくことにより，「和積形論理式の恒偽性」が判定できる．導出節を作る操作のことを「導出原理(resolution principle)」という．

  例: $g = (A \lor B) \land (A \lor \overline{B}) \land \overline{A}$

  $= (A \lor B) \land (A \lor \overline{B}) \land \overline{A} \land A$

  $= (A \lor B) \land (A \lor B) \land \overline{A} \land A \land \text{false}$

  $= \text{false}$

- 節(和項) t1 と t2 のリゾルベント(resolvent) res(ti, tj) とは：
  - ただ一組の変数だけ肯定と否定が含まれているとき，それを除いて，t1 と t2 に含まれるリテラル(命題記号またはその否定)の ∨ をとったもの(重複するリテラルは一つにまとめる).

    (リテラルp と¬p は相補的であるともいう．)
  - $\neg u \lor w \lor x \lor \neg y$ と $\neg u \lor w \lor x \lor y \lor z$ のリゾルベントは $\neg u \lor w \lor x \lor z$

和積形論理式 f の恒偽性判定

$t1 \land t2 \leq res(t1, t2)$

- 和積形論理式 f の恒偽性を，リゾルベント(導出節) をつくっていくことにより判定する方法:
- $f \equiv 1$ でないとする． $f = t1 \land t2 \land \cdots \land tn$ とする． 各ti は節(和項)で 0 でないとする．

(1). S を与えられた節集合{t1, t2, … , tn} に初期設定

(2). (a) S 中のどの ti, tj についても

– i. ti, tj のリゾルベントがない，か又は

– ii. 存在してもそのリゾルベント res(ti, tj) に対し，
$res(ti, tj) \geq tk$ なる tk が S にある をみたすならば終了，
f は恒偽でないと判定.

(b) そうでないとき，

– 「どの $tk \in S$ に対しても $res(ti, tj) \not\geq tk$」なる res(ti, tj) を一つ見つける．それが 0 なら終了，f は恒偽であると判定．0 でなければ，S に追加し，(2) をくりかえす．

なぜ否定の和積形の恒偽性を調べようとするのか

例1.2.1

– $P = (A \to \neg B) \land (\neg A \land B \to D) \land (\neg B \to \neg D) \land (\neg C \to \neg D)$
$\to (\neg A \land C \land D \lor \neg B \land \neg D)$
の恒真性を示せ．ただし，この否定が恒偽であることを，リゾルベントをつくっていく方法により示せ．

– 「…が恒偽であることを空節を導出することにより示せ」 というような言い方をすることもある．

– また，「…の否定が恒偽であることを」といちいち言わずに，「 …が恒真であることを導出原理を利用して示せ」などと言うこともある．

• 与式を P とおき，¬ P を和積形に直してみよう！

例1.2.1

- 与式 $P=(A \to \neg B) \land (\neg A \land B \to D) \land (\neg B \to \neg D) \land (\neg C \to \neg D)$
  $\to (\neg A \land C \land D \lor \neg B \land \neg D)$
  を $(\alpha_1 \land \cdots \land \alpha_4 \to (\beta_1 \lor \beta_2))$ と考え，$\neg P$ を和積形に直すと，
  $\neg P \equiv \alpha_1 \land \cdots \land \alpha_4 \land \neg(\beta_1 \lor \beta_2)$
  $\neg P \equiv \quad (A \to \neg B)$
  $\land (\neg A \land B \to D)$
  $\land (\neg B \to \neg D)$
  $\land (\neg C \to \neg D)$
  $\land \neg(\neg A \land C \land D \lor \neg B \land \neg D)$
- したがって， $\neg P$ の和積形の各節は次のようになる．

$\neg A \lor \neg B$	(1)
$A \lor \neg B \lor D$	(2)
$B \lor \neg D$	(3)
$C \lor \neg D$	(4)
$A \lor \neg C \lor \neg D$	(5)
$B \lor D$	(6)

例1.2.1

- リゾルベントを求めていく(例えば)

¬A ∨ ¬B	(1)
A ∨ ¬B ∨ D	(2)
B ∨ ¬D	(3)
C ∨ ¬D	(4)
A ∨ ¬C ∨ ¬D	(5)
B ∨ D	(6)
(3) と(6) より B	(7)
(1) と(7) より ¬A	(8)
(2) と(7) より A ∨ D	(9)
(8) と(9) より D	(10)
(4) と(10) より C	(11)
(5) と(8) より ¬C ∨ ¬D	(12)
(11) と(12) より ¬D	(13)
(10) と(13) より 0 (敢えて長いものを挙げておく)	

- ゆえに ¬P は恒偽．したがって，与式 P は恒真．

例1.2.1

- リゾルベントを求めていく(例えば)

¬A ∨ ¬B	(1)
A ∨ ¬B ∨ D	(2)
B ∨ ¬D	(3)
C ∨ ¬D	(4)
A ∨ ¬C ∨ ¬D	(5)
B ∨ D	(6)
(3) と(6) より B	(7)
(1) と(7) より ¬A	(8)
(2) と(7) より A ∨ D	(9)
(8) と(9) より D	(10)
(4) と(10) より C	(11)
(5) と(8) より ¬C ∨ ¬D	(12)
(11) と(12) より ¬D	(13)
(10) と(13) より 0 (敢えて長いものを挙げておく)	

- ゆえに ¬P は恒偽．したがって，与式 P は恒真．

例1.2.1

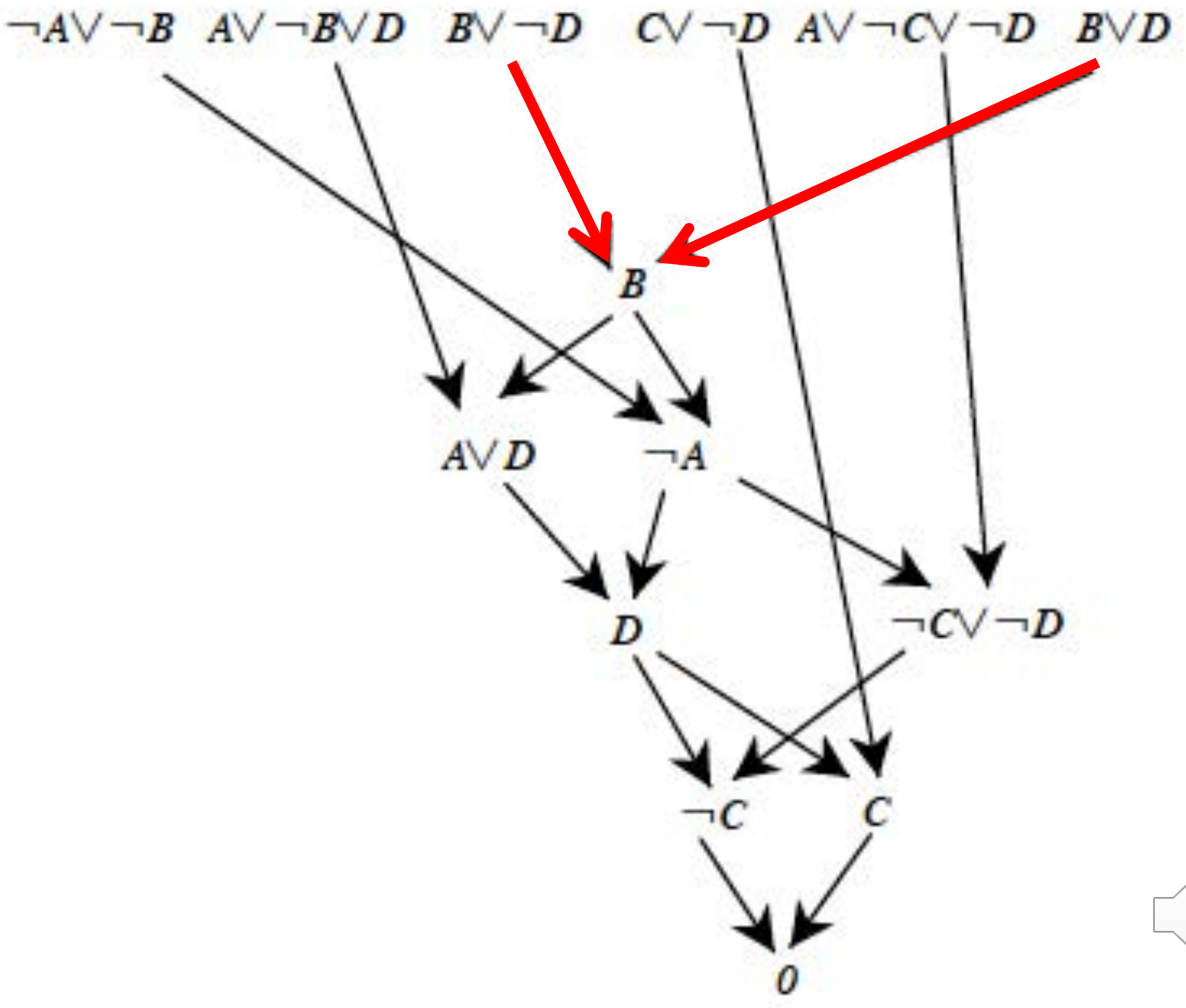
- リゾルベントを求めていく(例えば)

¬A ∨ ¬B	(1)
A ∨ ¬B ∨ D	(2)
B ∨ ¬D	(3)
C ∨ ¬D	(4)
A ∨ ¬C ∨ ¬D	(5)
B ∨ D	(6)
(3) と(6) より B	(7)
(1) と(7) より ¬A	(8)
(2) と(7) より A ∨ D	(9)
(8) と(9) より D	(10)
(4) と(10) より C	(11)
(5) と(8) より ¬C ∨ ¬D	(12)
(11) と(12) より ¬D	(13)

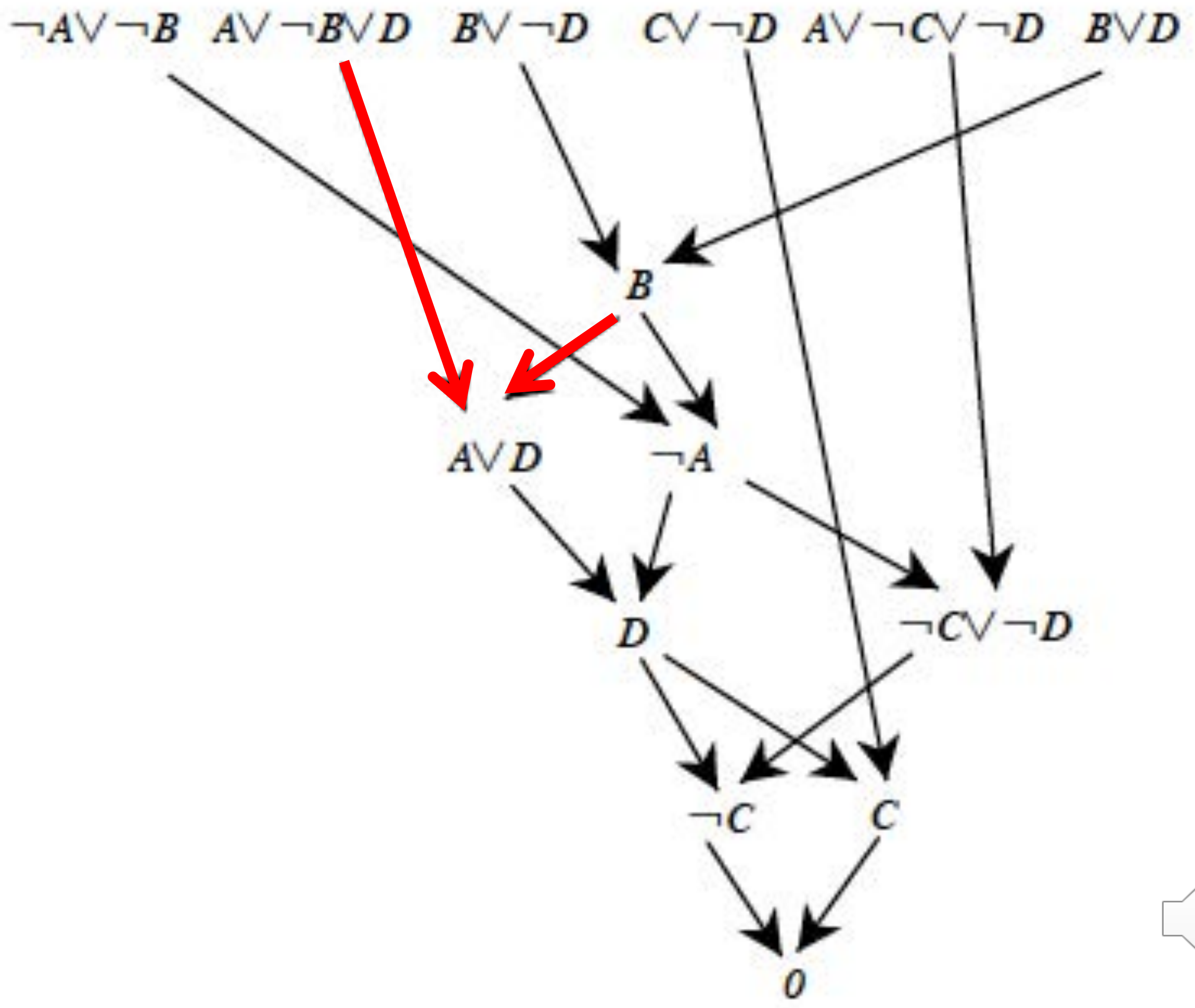
(10) と(13) より 0 (敢えて長いものを挙げておく)

- ゆえに ¬P は恒偽．したがって，与式 P は恒真．

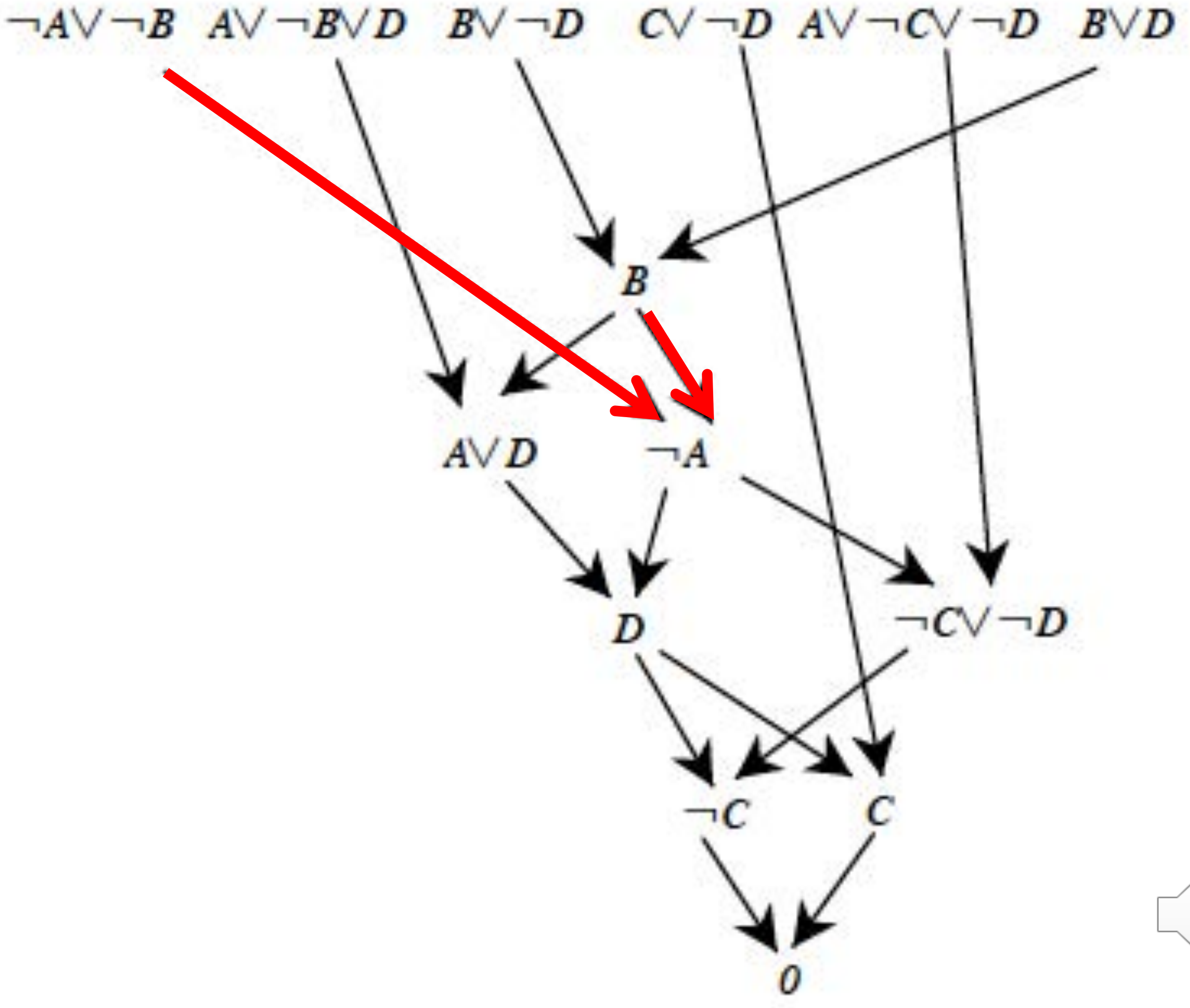
節0の導出過程を表す図



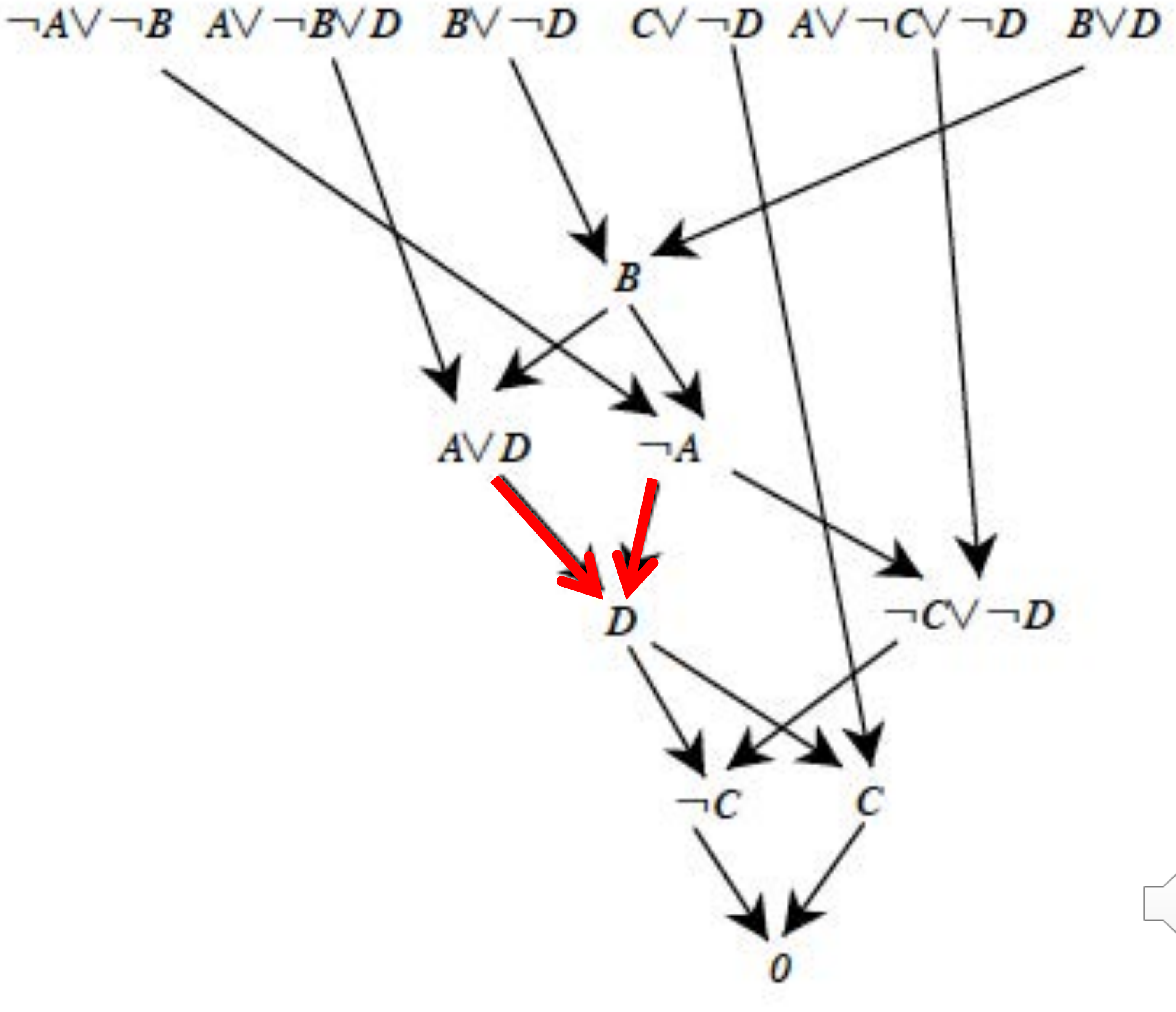
節0の導出過程を表す図



節0の導出過程を表す図

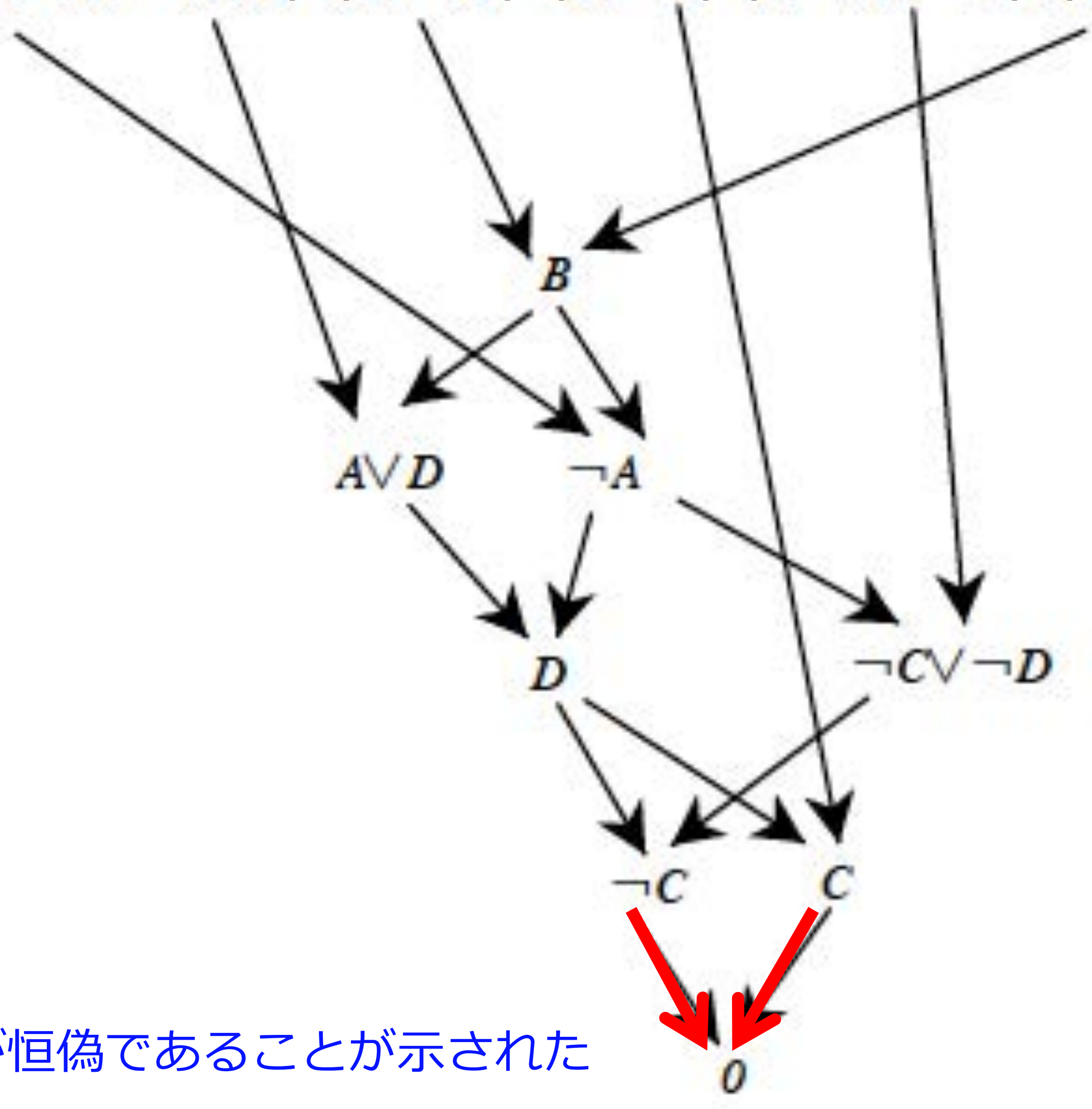


節0の導出過程を表す図



節0の導出過程を表す図

$\neg P = (\neg A \lor \neg B) \land (A \lor \neg B \lor D) \land (B \lor \neg D) \land (C \lor \neg D) \land (A \lor \neg C \lor \neg D) \land (B \lor D)$



¬Pが恒偽であることが示された

なぜ否定の和積形の恒偽性を調べようとするのか

- 論理式 R に対し「 ¬R を和積形で表し，リゾルベントをとっていって空節 0 を導出する」ことによって，R の恒真性を示すやり方は，「背理法 (帰謬(きびゅう) 法)」と見なせる.
  - ¬R を仮定する．そうすると， ¬R の和積形の各節は真.
  - 節 ti，tj が真なら res(ti, tj) も真でなければならない (ti ∧ tj ≤res(ti, tj) より).
  - したがって，もとの節や出てきたリゾルベントの節は全て真.
  - ある命題記号 p に対し，節 p と ¬p が得られたなら，矛盾.
  - リゾルベント法では res(p,¬p) = 0 を求めて終る.
  - よって，R は成り立つ(恒真) と結論する.
- P ≡ (和項t1∧…∧和項tn) → (積項t’1∨… ∨積項t’m) なら，

$$\neg P \equiv (\text{和項}t1 \wedge \cdots \wedge \text{和項}tn) \wedge \overline{(\text{積項}t'1} \wedge \cdots \wedge \overline{\text{積項}t'm)}$$

$$\equiv \text{和項}t1 \wedge \cdots \wedge \text{和項}tn \wedge \text{和項}t''1 \wedge \cdots \wedge \text{和項}t''m \quad (t''j=\overline{t'j})$$

- もし ¬P≡0 が示せれば，P≡1 が示せる.
- 和項 tj が (和項t1∧…∧和項tn) → (積項t’1∨…∨積項t’m) でも良い.

人工知能の分野の論理式

- 命題記号及びそれに否定がついたものをリテラルと呼ぶ．人工知能の分野では，次の形の論理式の恒真性を問うことが多い．
- $P1 \land P2 \land \cdots \land Pn \rightarrow Q$
- Pi は
  - (1). 一つのリテラル，又は
  - (2). いくつかのリテラルの論理積 → リテラル
- ((1) は前提の事実，(2) は推論の知識)
  - Q はいくつかのリテラルの論理積
- この形なら，この否定の恒偽性を，リゾルベント(導出節) をつくっていくことにより調べるのが得策．

この辺の詳しい話は３年生配当の情報論理学の中川先生の授業で聞いてください

問1.2.8

- 次のことは正しいか．命題論理式で表して調べよ．

  A 点が高圧なら，C 点の流量は多い．

  C 点が高圧でなく，かつB 点の流量が多くなければ，A 点は高圧でない．

  A 点，B 点，C 点がすべて高圧なら，C 点が異常である．

  C 点が高圧で，かつC 点の流量が多ければ，B 点が異常である．

  A 点が高圧である．

  B 点の流量は大でない．

  そうなら，B 点が異常であるか?

ヒント:

- 論理的に正しいかを調べるには，論理式で表して，その論理式が恒真であることを確かめれば十分である．A 点が高圧であることを命題記号 p_A で，B 点が高圧であることを p_B で，C 点が高圧であることを p_C で，B 点の流量が多いことを s_B で，C 点の流量が多いことを s_C で，B 点が異常であることを t_B で，C 点が異常であることを t_C で表すと，上の文はどのような論理式で表せるか考えてみよう．

問1.2.8

- 次のことは正しいか．命題論理式で表して調べよ．

  A 点が高圧なら，C 点の流量は多い．

  C 点が高圧でなく，かつB 点の流量が多くなければ，A 点は高圧でない．

  A 点，B 点，C 点がすべて高圧なら，C 点が異常である．

  C 点が高圧で，かつC 点の流量が多ければ，B 点が異常である．

  A 点が高圧である．

  B 点の流量は大でない．

  そうなら，B 点が異常であるか?

論理式:

$$
\begin{aligned}
&(p_A \to s_C)\\
&\wedge\ (\neg p_C \wedge \neg s_B \to \neg p_A)\\
&\wedge\ (p_A \wedge p_B \wedge p_C \to t_C)\\
&\wedge\ (p_C \wedge s_C \to t_B)\\
&\wedge\ p_A\\
&\wedge\ \neg s_B\\
&\quad \to t_B
\end{aligned}
$$

問1.2.8

- 次のことは正しいか．命題論理式で表して調べよ．

  論理式: $P \equiv \{前提\} \rightarrow 結論$

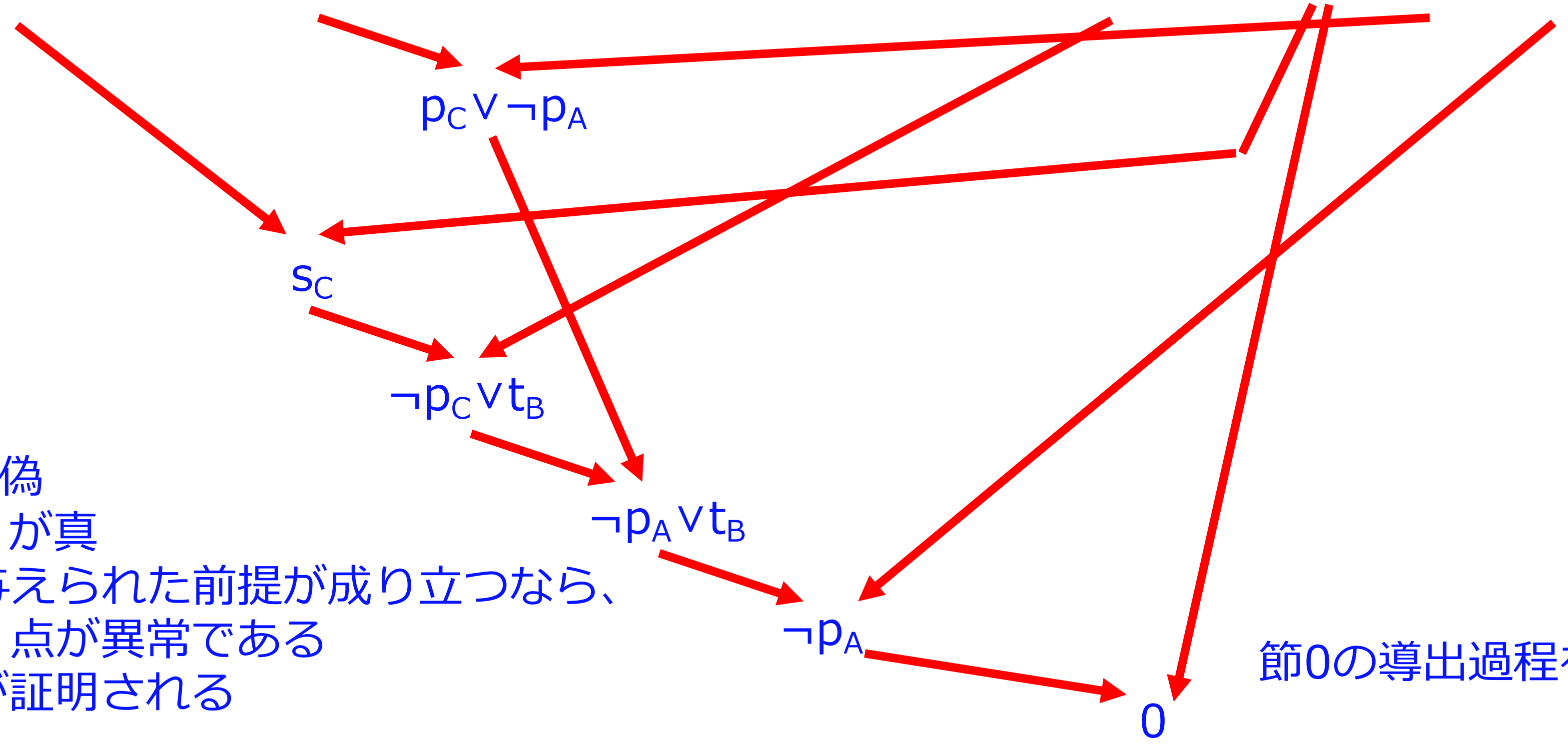
$\{ (p_A \rightarrow s_C)\wedge(\neg p_C\wedge\neg s_B \rightarrow \neg p_A)\wedge(p_A\wedge p_B\wedge p_C \rightarrow t_C)\wedge(p_C\wedge s_C \rightarrow t_B)\wedge p_A\wedge\neg s_B\} \rightarrow t_B$

  論理式: $\neg P \equiv \neg\{\{前提\} \rightarrow 結論\} \equiv 前提 \wedge \neg 結論$

$(p_A \rightarrow s_C)\wedge(\neg p_C\wedge\neg s_B \rightarrow \neg p_A)\wedge(p_A\wedge p_B\wedge p_C \rightarrow t_C)\wedge(p_C\wedge s_C \rightarrow t_B)\wedge p_A\wedge\neg s_B\wedge\neg t_B$

$=$

$(\neg p_A\vee s_C)\wedge(p_C\vee s_B\vee\neg p_A)\wedge(\neg p_A\vee\neg p_B\vee\neg p_C\vee t_C)\wedge(\neg p_C\vee\neg s_C\vee t_B)\wedge p_A\wedge\neg s_B\wedge\neg t_B$



$\neg P$ が偽
 ＝ P が真
 ＝ 与えられた前提が成り立つなら、
  B 点が異常である
ことが証明される

節0の導出過程を表す図

論理関数の使い方

- 論理関数は次のような目的に使われる
  - 論理設計
    - 組み合わせ論理回路の設計
    - 順序回路の設計
    - CPUの設計
  - 情報論理学
    - $P1 \land P2 \land \cdots \land Pn \rightarrow Q$ の形の論理式の恒真性判定
    - 人工知能の基礎
    - データベースの質問応答
    - パズルの解法
    - 組み合わせ問題
    - 等（詳しくは３年生の情報論理学の授業で聞いてください）

４回目の授業終了

授業終了

皆さん
今日はレポート課題はありません