

クワインマキラスキー法・ 論理式簡単化ソフトウェア 基本設計書

学籍番号：BP19094

氏名：阿部ブライアン

目次

1. コントロールとイベント設計	2
1-1. 画面レイアウト	2
1-2. 入力端子数設定	2
1-3. 入力パネル（加法標準形入力）	3
1-4. 真理値表入力	4
1-5. 簡単化された論理式の導出	4
2. メソッド設計	5
2-1. void InitInputPanel() ... 入力パネルの生成	5
2-2. void InitPatternTable() ... 真理値表を生成	5
2-3. void UpdateSetText() ... 入力中の項の更新	5
2-4. void CalcValue() ... 項情報の計算	5
2-5. void SetOut(int n) ... 真理値表の出力を設定	5
2-6. Boolean CanUse(InSet inSet) ... 加法標準形にその要素があるか検索	5
2-7. Boolean Comparison(InSet inSet1, InSet inSet2) ... 2つの項が合成可能か検索	6
2-8. InSet Merge(InSet inSet1, InSet inSet2) ... 2つの項を合成し、結果を返す	6
2-9. List<InSet> DeleteSame() ... 重複する項を削除し、削除後のリストを返す	6
2-10. void CalcUnique() ... 各項の持つ唯一なパターンを数え、主項順に重みづけ	6
2-11. void DisplayAns() ... 簡単化の結果を表示する	6
2-12. void OutputDebug(string str) ... エラー表示をする	6
2-13. void CleanSet() ... 主項の検討を行い、不要な項を削除する	7

1. コントロールとイベント設計

1-1. 画面レイアウト

図 1 に実際の動作画面を示す。

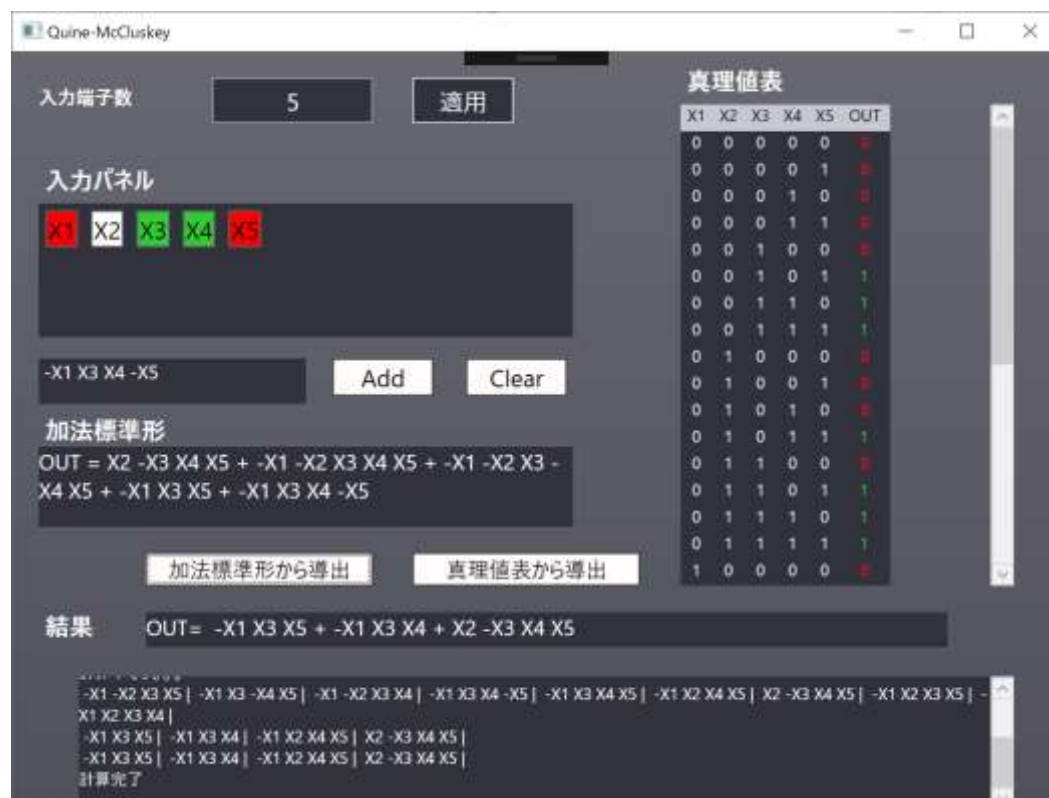


図 1 画面レイアウト

各コントロールの役割と呼び出すイベントを以下より示す。

1-2. 入力端子数設定



図 2 入力端子数設定コントロール

- ① 「適用」 ボタンをクリック => void UpdateInputCount();

役割

左の TextBox の値に入力端子数を更新し、入力端子数に応じて、入力パネルと真理値表を作成する。

例外対応

- 数字以外を入力 ⇒操作を取り消して、エラー表示
- 2 未満の値を入力 ⇒操作を取り消して、エラー表示

下位メソッド

- void InitInputPanel() … 入力パネルの生成
- void InitPatternTable() … 真理値表の生成
- void UpdateSetText() … 現在入力中の項の表示更新

1-3. 入力パネル（加法標準形入力）



図 3 入力パネル

- ① 各入力要素ボタンをクリック => void Button_Click();

役割

入力要素の値を切り替え

無(白) → 真(緑) → 偽(赤) という順番で切替

図 3 左下の現在の項の情報を更新する

下位メソッド

- void UpdateSetText() … 現在入力中の項の表示更新

- ② 「Add」 ボタンをクリック => void Add_Button_Click();

役割

現在の項を加法標準形に追加

例外対応

- すでに追加されている項を入力 ⇒操作を取り消して、エラー表示

- ③ 「Clear」 ボタンをクリック => void Clear_Button_Click();

役割

現在入力されている加法標準形をクリア

1-4. 真理値表入力



X1	X2	X3	X4	X5	OUT
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0

図 4 真理値表コントロール

- ① OUT を値(数字)クリック => void ChangeOut();

役割

真理値表の出力を切り替える 0(赤) ⇔ 1(緑)

1-5. 簡単化された論理式の導出

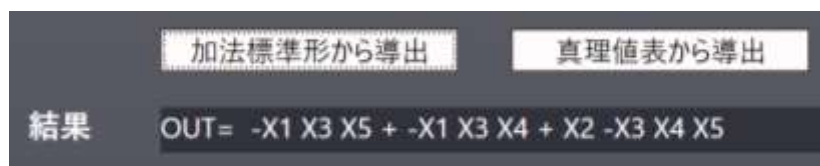


図 5 導出表示

- ① 「加法標準形から導出」 ボタンをクリック => void Calc_Form_Button_Click();

役割

加法標準形から、真理値表を導出し、それを真理値表コントロールに反映

真理値表からクワインマクラスキー法を実行し簡単化を行い、結果の TextBox に表示。

例外対応

- 空の状態をクリック ⇒そのまま実行し、空が出力される

下位メソッド

- void InitPatternTable() … 真理値表の生成(初期化)
- void Calc() … クワインマクラスキー法で簡単化を行い表示。

- ② 「加法標準形から導出」 ボタンをクリック => void Calc_Form_Button_Click();

役割

真理値表からクワインマクラスキー法を実行し簡単化を行い、結果の TextBox に表示。

下位メソッド

- void Calc() … クワインマクラスキー法で簡単化を行い表示。

2. メソッド設計

2-1. void InitInputPanel() … 入力パネルの生成

- ① リスト・入力パネル、加法標準形を初期化
- ② 入力端子数分の Input を用意し、リストに追加
- ③ 生成したボタンをコントロールに追加

2-2. void InitPatternTable() … 真理値表を生成

- ① リスト・真理値表を初期化
- ② 入力端子数+1 分の Header を用意し、リストに追加
- ③ ヘッダー部分を真理値表に追加
- ④ 入力端子数+1 分の Pattern を用意し、リストに追加
- ⑤ Pattern に入力パターン (2 進数で $0 \sim 2^{\text{input_count}}$) を設定
- ⑥ Pattern 行をヘッダーサイズに合わせて、真理値表に追加

2-3. void UpdateSetText() … 入力中の項の更新

- ① 入力中の項を初期化
- ② ステータスが無以外のボタンの入力値を追加

2-4. void CalcValue() … 項情報の計算

- ① 項の入力値の 10 進数値と 1 入力 の数を初期化
- ② 入力値から入力値の 10 進数表記を計算
- ③ 入力値から 1 の入力 の数をカウント

2-5. void SetOut(int n) … 真理値表の出力を設定

- ① 出力を引数 n に設定
- ② 1 ならば緑色にする
- ③ 0 ならば赤色にする

2-6. Boolean CanUse(InSet inSet) … 加法標準形にその要素があるか検索

- ① 加法標準形のリストから inSet と同じものを検索
- ② 検索結果が null であれば True、そうでなければ False を返す

2-7. Boolean Comparison(InSet inSet1, InSet inSet2) … 2つの項が合成可能か検索

- ① inSet1 と inSet2 の 1 入力の数之差が 1 でなければ False を返す
- ② inSet1 と inSet2 の入力値を比較し、異なる場所を数える
- ③ 異なる場所が 1 つであったら True を返す (合成可能)
- ④ そうでなければ False を返す

2-8. InSet Merge(InSet inSet1, InSet inSet2) … 2つの項を合成し、結果を返す

- ① 新しい InSet after を作成する
- ② inSet1 と inSet2 の入力値を順番に比較する
- ③ 入力値が同じ場所は、after にその入力値を反映させる。
- ④ 入力値が異なる場所は、after にそのドントケアを反映させる。
- ⑤ after を返す

2-9. List<InSet> DeleteSame() … 重複する項を削除し、削除後のリストを返す

- ① 新しいリストを用意
- ② 既存のリスト内の項を一つずつ参照する
- ③ 参照している項と同じものが別に存在するか、既存リストから検索する
- ④ 同じものが見つからなかったら、その項を新しいリストに追加する
- ⑤ 新しいリストを返す

2-10. void CalcUnique() … 各項の持つ唯一なパターンを数え、主項順に重みづけ

- ① 各項のパターンリスト参照し、検討パターンを定める
- ② 他の項のパターンリストを参照する
- ③ 他の項のパターンリストから検討パターンと同じものがあるか検索する
- ④ 同じものが見つからなかったら、その項の unique を 1 加算する

2-11. void DisplayAns() … 簡単化の結果を表示する

- ① 表示を初期化する
- ② 各項を表示に追加する

2-12. void OutputDebug(string str) … エラー表示をする

- ① エラー表示に str を追加し、下までスクロールする。

2-13. void CleanSet() … 主項の検討を行い、不要な項を削除する

- ① 項のリストの重複を無くす
- ② CalcUnique0を呼び出す … 各項の持つ唯一なパターンを数え、主項順に重みづけ
- ③ 各項を参照する
- ④ Boolean used を false で定義 ユニークな入力を持つか否か
- ⑤ 参照中の項の、入力値のリストを参照
- ⑥ 項の入力値を、全体の List_value から検索
- ⑦ 検索結果が見つからなかった場合は、その値を全体の List_value から消去し、used=trueとする
- ⑧ used を各項の used に代入する。

これによって used が false の物はユニークな入力を持たず、不要な項ということになる。