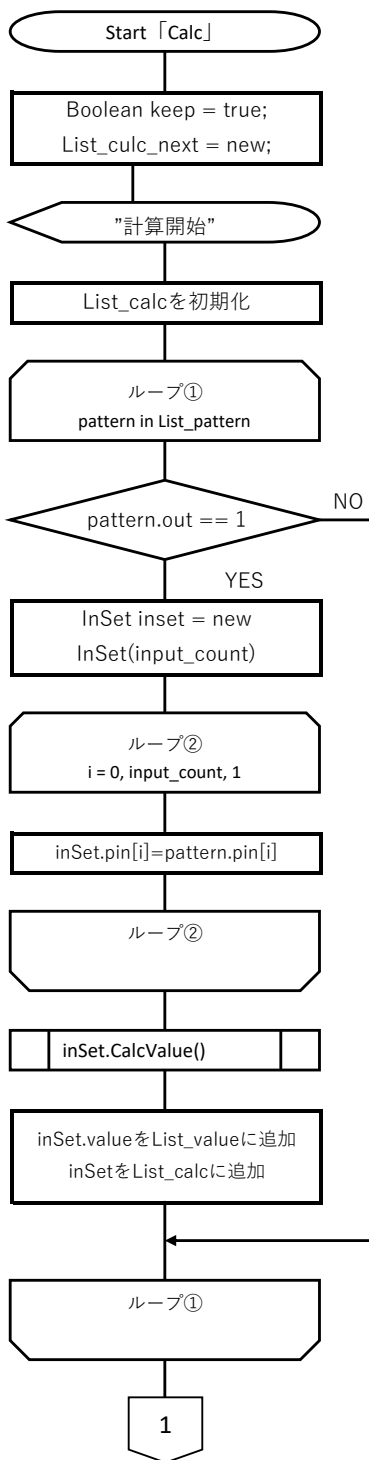


クワインマキラスキー法 Calcメソッド 詳細設計

BP19094

阿部ブライヤン



keep : 計算継続の判断用

List_culc_next : 計算結果の項の格納用リスト

//パターンごとの項の導出開始

List_calc : 計算に使用する項の入ったリスト

ループ① : List_patternの要素patternを順に参照

出力値が1であれば

//項の導出開始

新しい項を宣言

ループ② : 変数iを0からinput_countまで1ずつ加算

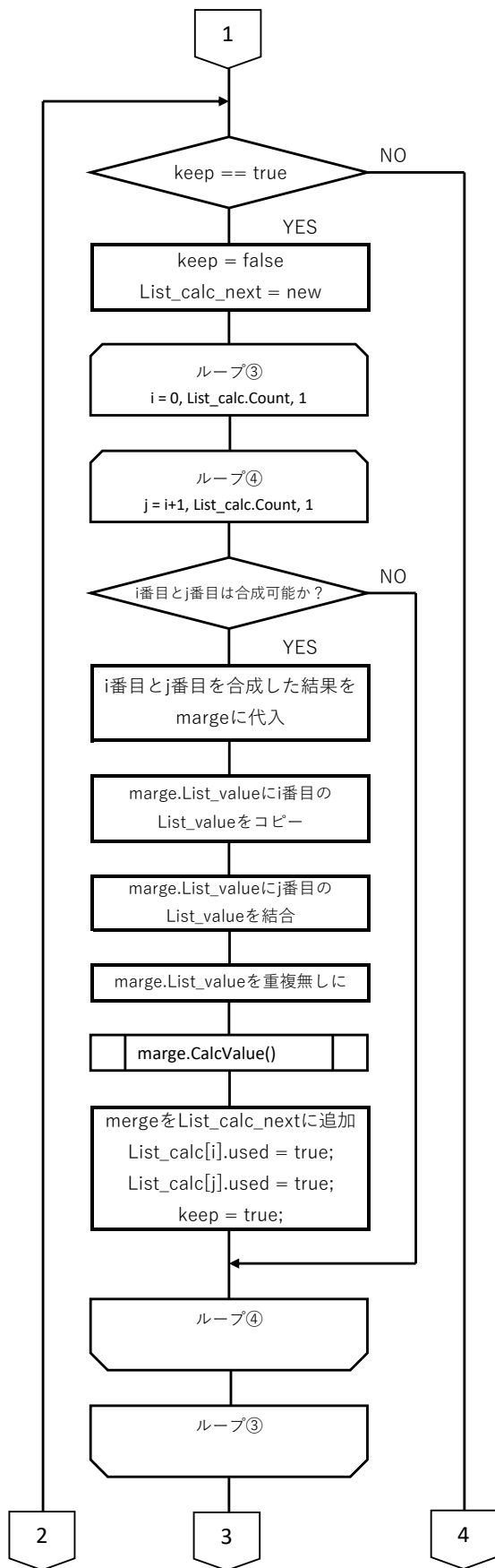
入力情報のコピー

ループ②終了

void CalcValue() … 項情報の計算

List_value : 入力値の10進表記。主項の検討に用いる

ループ①終了



継続判定が出ている間ループ

keepとList_calc_nextを初期化

ループ③：変数iを0からList_calcの要素数まで1ずつ加算

ループ④：変数jをi+1からList_calcの要素数まで1ずつ加算

Comparison(List_calc[i], List_calc[j])

Boolean Comparison(InSet, InSet) … 2つの項が合成可能か検索

InSet marge = Merge(List_calc[i], List_calc[j])

InSet Merge(InSet, InSet) … 2つの項を合成し、結果を返す

merge.List_value = List_calc[i].List_value;

merge.List_value.AddRange(List_calc[j].List_value);

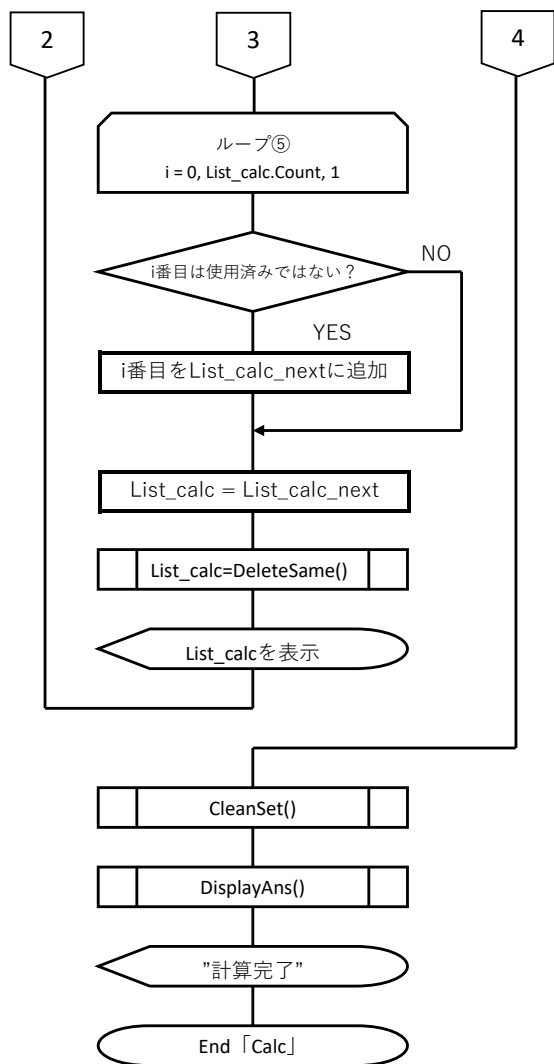
merge.List_value = merge.List_value.Distinct().ToList();

合成に使った項を使用済みに変更

継続条件をtrueに

ループ④終了

ループ③終了



//合成に使われなかった項目の継承

ループ⑤：変数iを0からList_calcの要素数まで1ずつ加算

```
if (List_calc[i].used == false)
```

```
List_calc_next.Add(List_calc[i]);
```

計算結果を反映

List<InSet> DeleteSame() …重複する項を削除し、リストを返す

計算途中の状況を表示

void CleanSet() … 主項の検討を行い、不要な項を削除する

void DisplayAns() … 簡単化の結果を表示する