リバースモデリングツール

RExSTM for C

ユーザマニュアル

改訂履歴

版数	発行日	ツールバージョン	改定者	改定内容
第 1 版	2018.07.17	Ver2.0.0	清水貴裕[1]	初版作成
第 2 版	2019.05.30	Ver2.1.0	清水貴裕[1]	5.5 章を追記

[1]名古屋大学

目次

1. ツールの目的

本ツールの目的は、C 言語のソースコードから状態遷移表をリバース生成することである。

2. ディレクトリ構造

本ツールのディレクトリ構造について、ツールのルートディレクトリ（以降、ツールのルートディレクトリを RExSTM と表記する）は、図 1 のようになっている。

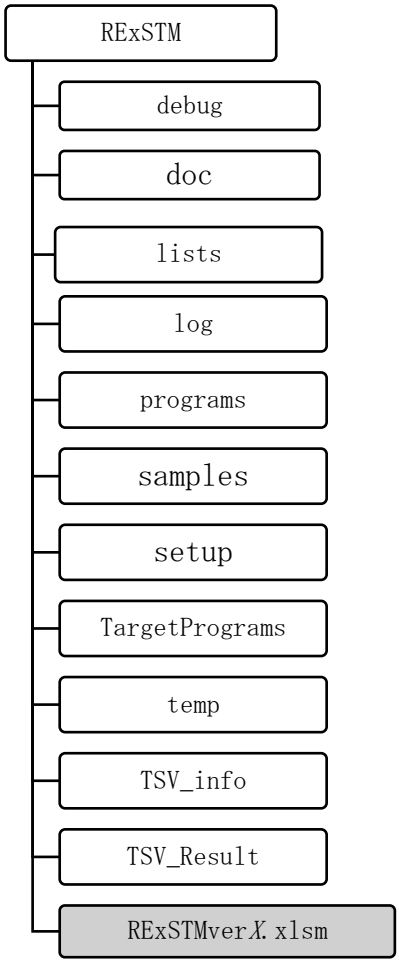

図 1 ディレクトリ構造（*X* はバージョンの値を表す）

ユーザが利用するディレクトリおよびファイルについて説明する。

- ディレクトリ TargetPrograms には、ユーザが解析対象とするソースファイルを置く。
- ディレクトリ lists には、設定ファイルがある。詳細は 4.2 章の (2) にて説明する。
- RExSTMver*X*.xlsm は、ソースコードの解析、条件処理表の作成、状態遷移表の作成をするマクロ付 Excel ファイルである。

3. 実行環境

本ツールでは、以下の環境での動作を想定している。

- Windows7 以降(Windows7 32bit/64bit, Windows10 64bit にて動作確認済)
- Microsoft Office Excel 2010 以降(2010, 2013, 2016 にて動作確認済)
- MinGW gcc(gcc version 6.3.0 (MinGW.org GCC-6.3.0-1) にて動作確認済)

MinGW gcc のインストールについては、別紙の「gcc インストールガイド」を参照すること。

3.1. コンパイルオプション

RExSTM/lists/compile.txt にデフォルトのコンパイルオプションが記載されている。compile.txt を書き換えることでコンパイルオプションを変更することができる。デフォルトでは、本ツールの動作に必要となるマクロ展開およびコメント削除を行うためのオプションを設定している。オプションの削除はせず、追加のみ行うこと。

3.2. コンパイラの変更について

動作保証外だが、コンパイラを変更することができる。現在、本ツールでは MinGW gcc のみをサポートしており、使用するコンパイラを変更する場合は、新たに指定したコンパイラによって本ツールが動作しない可能性や予期せぬ動作をする可能性を考慮の上、自己責任で実施すること。

RExSTM/lists/compile.txt にデフォルトのコンパイラが記載されている。compile.txt を書き換えることでコンパイラを変更することができる。

4. ツールの使用方法

本ツールの使用方法を以下に示す。本説明は Microsoft Office 2016 を想定している。

4.1. 事前準備

ツールの使用の前に以下の作業を行う必要がある。

(1) ビルド

ツールの初回利用時、および、ツールの変更を行った際にはビルドが必要である。ビルド方法については RExSTM/doc 内の「ビルド用ドキュメント」を参照すること。

(2) ファイルの準備

RExSTM/TargetPrograms に解析対象のソースファイル(*.c) とインクルードするヘッダファイルを置く。ヘッダファイルに関して、ソースファイル内のプリプロセッサ命令 #include 文が相対アドレスを示している場合は、ディレクトリ TargetPrograms がカレントディレクトリであることとして、その相対アドレスのとおりの場所に保存する。

なお、対象ソースファイルについて、ファイルごとに条件処理表の作成および状態遷移表の作成が行われる。

(3) マクロの有効の確認

Excel ファイル(RExSTMver*X*.xlsm)を開く。開いたとき、マクロが有効になっていることを確認する (2 度目以降開く場合、マクロを無効にしていなければこの手順をスキップしてよい)。Excel の「ファイル」タブ-> 「オプション」タブ -> 「セキュリティセンター」タブ -> 「セキュリティセンターの設定」ボタン と進めていくと、図 2 のウィンドウが表示される。図 2 の赤枠 1 のマクロの設定を選択し、赤枠 2 の「すべてのマクロを有効にする」が選択されてい

れば、マクロは有効になっている。

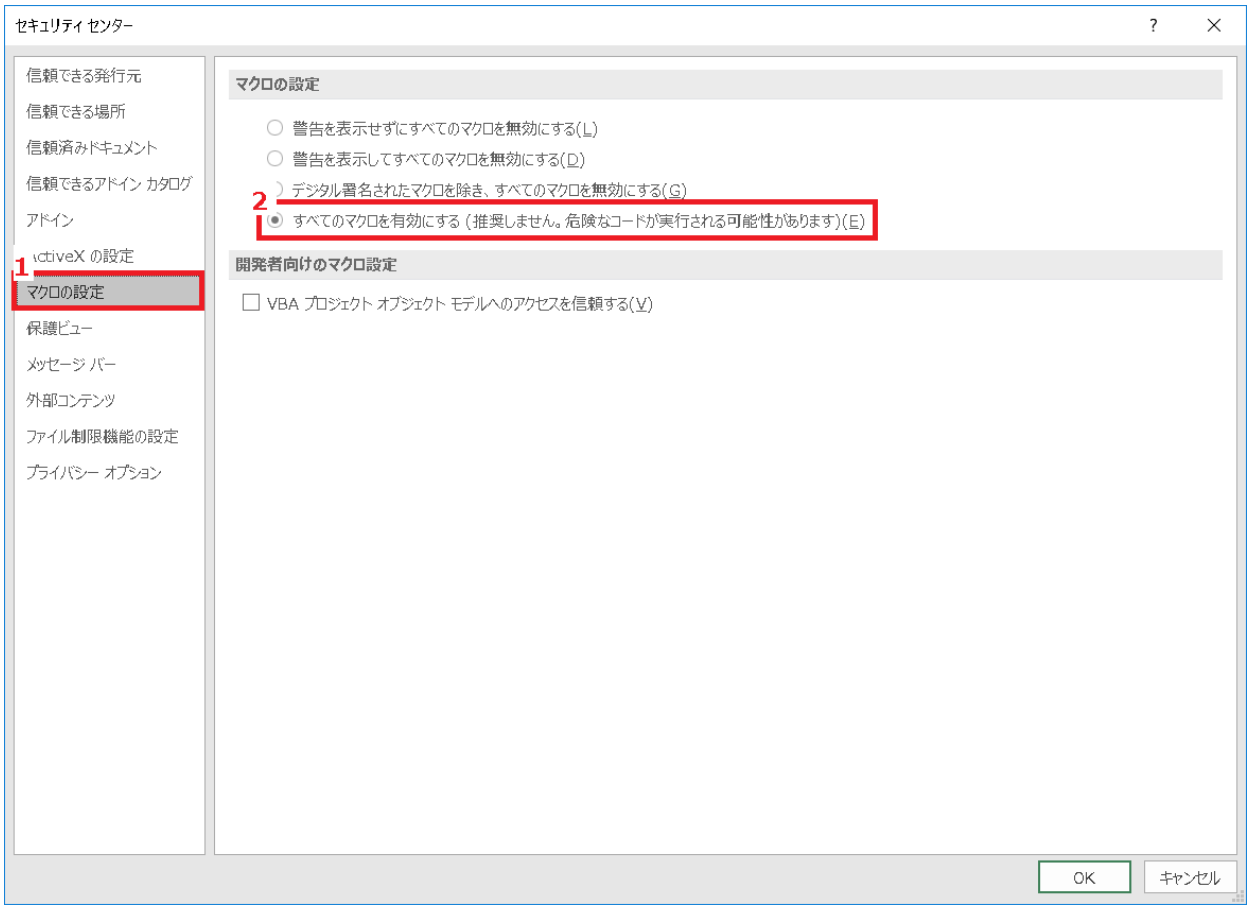

図 2 マクロが有効になっていることの確認

4.2. 条件処理表の作成

事前準備が終わったら、条件処理表を作成する。

(1) 条件処理表を作成開始する

Excel 上部のリボンから「RExSTM」を選択し（図 3 赤枠 1)、「条件処理表作成」ボタン（図 3 赤枠 2)を選択すると、ダイアログボックスが開き、図 4 が表示される。以降、RExSTM/TargetPrograms に存在するソースファイル全てに対して一括で処理される。また、同名のソースファイルを入力とした場合、古い条件処理表が新しいものに置き換わる。

これ以降の手順で状態遷移表を作成できない場合、RExSTM/temp ディレクトリのファイルを削除し、条件処理表の作成から再実行を行うことで解決される場合がある。

図 3 「条件処理表作成」ボタン

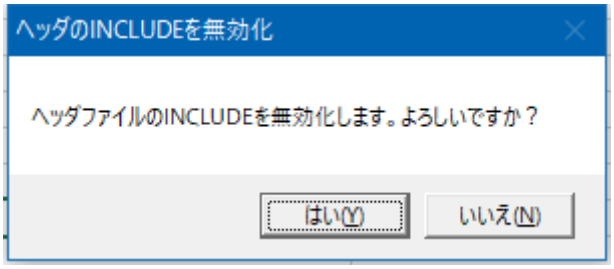

図 4 ヘッダファイルのインクルードの確認

(2) インクルードファイルの決定

図 4 でプリプロセッサ命令 #include 文を完全に無視して解析するならば「はい(Y)」を、#include 文を一部使用する場合は「いいえ(N)」を選択する。

「はい」、「いいえ」をそれぞれ選択した場合にソースファイルから削除される #include 文とマクロ展開される #include 文をまとめたものが表 1 である。

表 1 解析されるヘッダファイルについて

選択	「はい」	「いいえ」
ソースコードから削除される #include 文	すべて	no_included_files.txt と std.txt に記述されたもの
マクロ展開される #include 文	なし	上記以外

「はい」を選択した場合には #include 文はすべて削除され、「いいえ」を選択した場合には、RExSTM/lists/std.txt と RExSTM/lists/no_included_files.txt に記述されているヘッダファイルに関する #include 文のみ削除され、そのほかの #include 文を gcc によってマクロ展開する。std.txt には標準ライブラリが記述されており、no_included_files.txt にはユーザがマクロ展開したくないヘッダファイルを記述する。

（注意）「条件処理表作成」ボタンを選択すると、RExSTM/TargetPrograms に存在するソース

ファイル全てに対する条件処理表が作成されるが、図 4 のダイアログボックスは一度だけ表示され、すべての対象ソースファイルに同様の処理を行うものである。

(3) 条件処理表出力完了

(1)(2) の手順を実行すると、ツールが自動的に新規の Excel シートを生成し、条件処理表を出力する。生成されるシート名は“CP_【ソースファイル名】”であり、メイン操作画面シートの右側に生成される。

また、ソースファイルの構文解析結果は RExSTM/TSV_Result に TSV ファイルとして出力される。既に同名のファイルが存在する場合は上書きされる。

（注意）条件処理表出力後、RExSTM/TargetPrograms 内のソースファイルを変更して再度条件処理表を出力する場合は、再出力の前に 4.4 章の「シート削除」を行うこと。

4.3. 状態変数の選択と状態遷移表の作成

条件処理表を作成後に、状態変数の選択と状態遷移表の作成を行う。

Excel のリボン「RExSTM」（図 5 赤枠 1）から、「状態変数選択フォーム」ボタンを選択すると（図 5 赤枠 2)、状態変数選択フォームが表示される。このフォーム上で状態変数を選択する（状態変数の候補として表示される変数については「5.5 状態変数についての制約」を参照すること)。

状態変数選択フォーム上で状態変数を選択した後、「状態遷移表の生成」ボタンを選択すると、状態遷移表が作成される。

また、同名の状態変数を選択して状態遷移表を作成した場合、古い状態遷移表が新しいものに置き換わる。今回のリリース版では、別ソースファイルであっても同名の状態変数を選択した場合、状態遷移表が上書きされる。

図 5 「状態変数選択フォーム」ボタン

状態変数選択フォームの使用方法を説明する。

(1) 対象ソースファイルの選択

図 6 の赤枠内のコンボボックスより対象とするソースファイルを選択する。選択可能なソースファイルは、すでに条件処理表が作成されているソースファイルのみである。

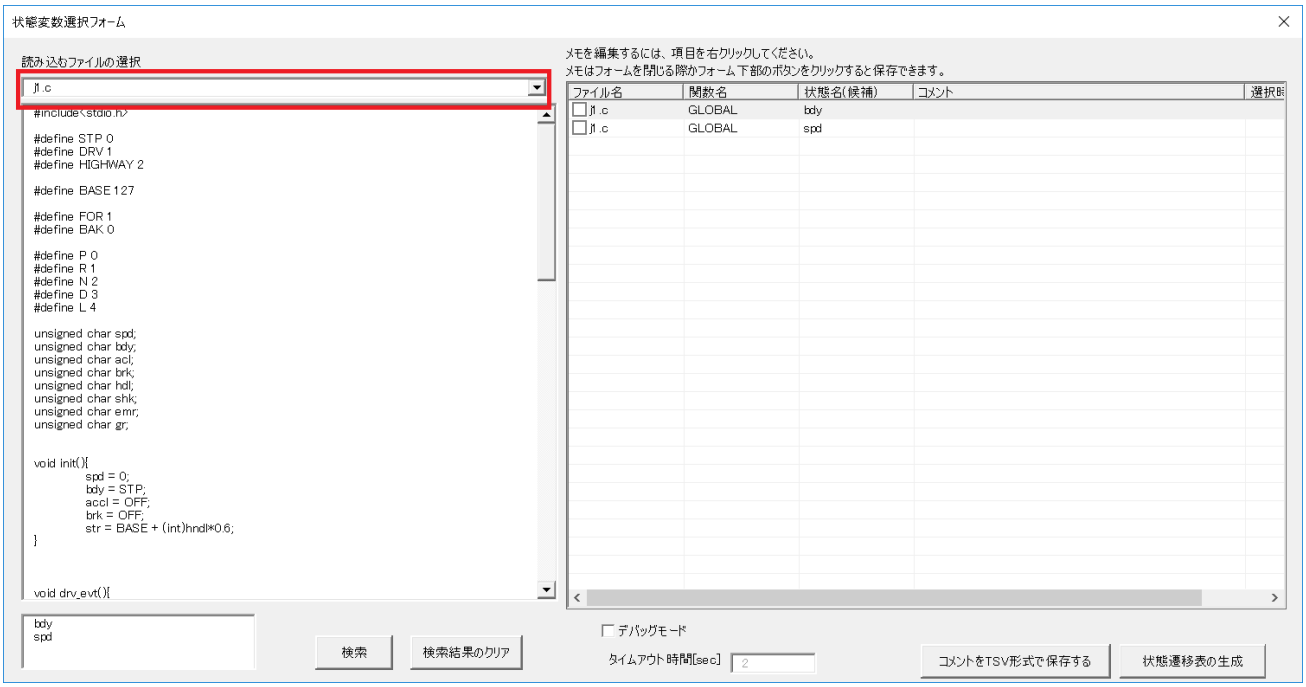

図 6 対象ソースファイルの選択

(2) 対象ソースファイル内における状態変数のハイライト機能

ソースファイルの選択後、図 7 の赤枠 1 において、リストボックスから検索する変数を選択後、赤枠 2 の「検索」ボタンを押下すると選択した変数を含む行がハイライトされる。

また、赤枠 3 の「検索結果のクリア」ボタンを押下すると、ハイライトがクリアされる。

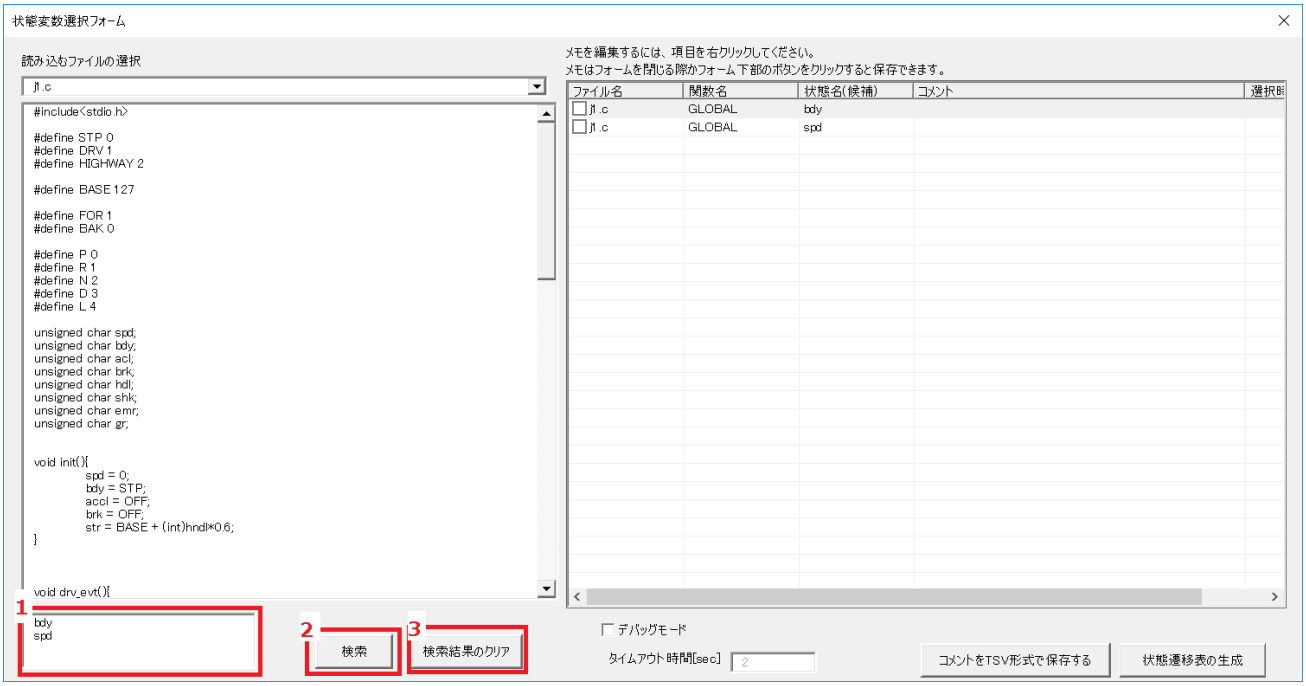

図 7 コメント機能

(3) 状態変数に対するコメント機能

図 8 の赤枠 1 のコメント列において右クリックすることで、コメントの入力ができる。コメントは、赤枠 2 の「コメントを TSV 形式で保存する」ボタンを選択するか、フォームを閉じることで RExSTM/log に TSV ファイルとして保存される。

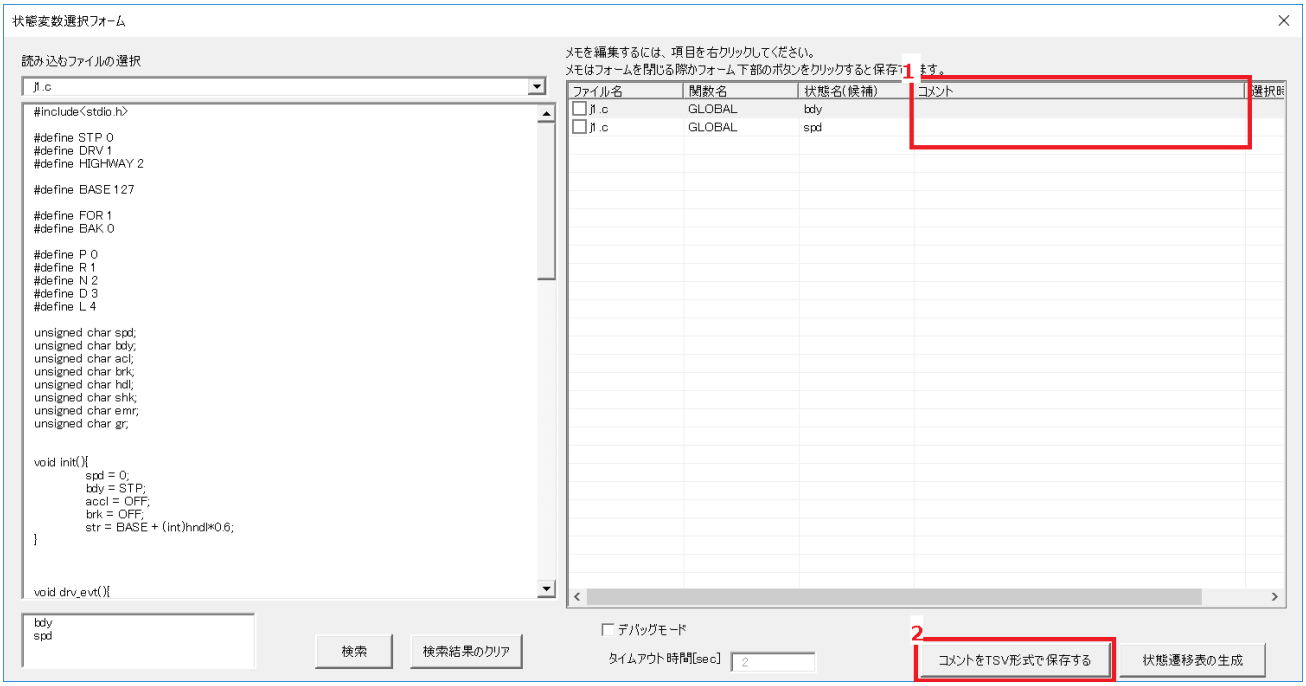

図 8 コメント機能

(4) 状態遷移表の出力

図 9 の赤枠 1 で対象としたい状態変数のチェックボックスにチェックを入れ、赤枠 2 の「状態遷移表の生成」ボタンを選択すると、対象とした状態変数に対する状態遷移表が出力される。ただし、本ツールでは状態変数の選択はただ 1 つのみをサポートする。

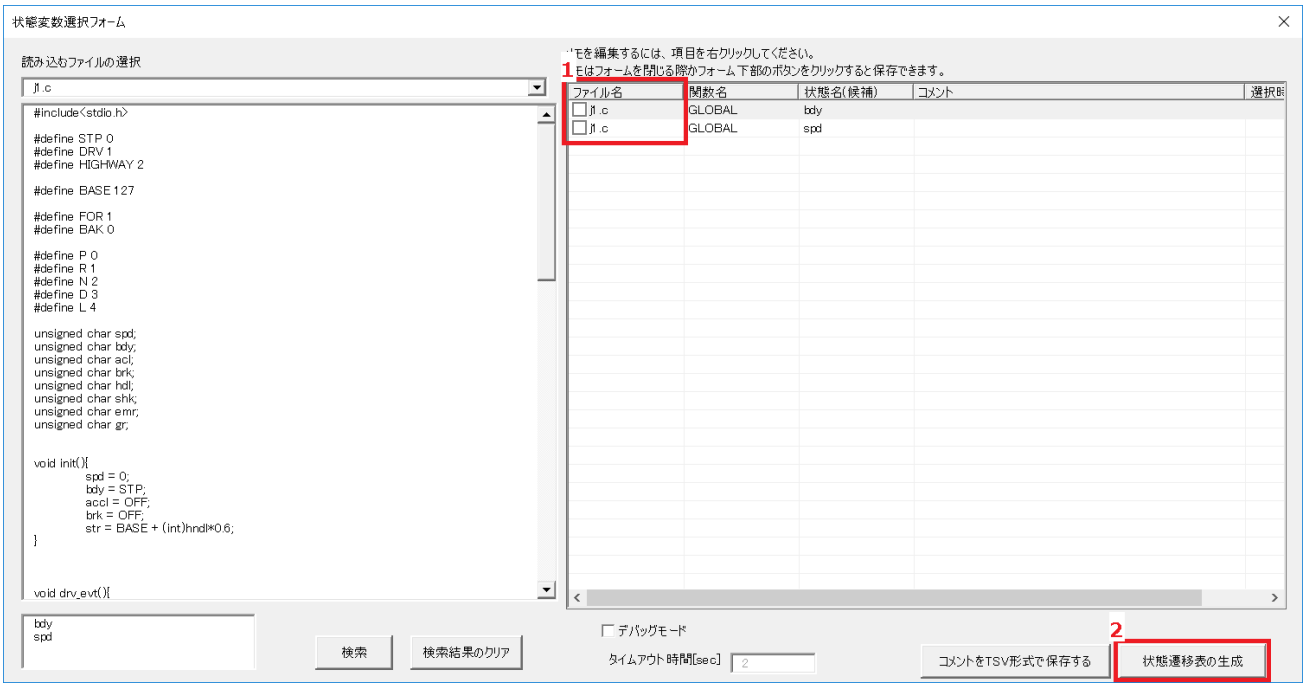

図 9 状態遷移表の出力

4.4. シートの削除

シートを削除する方法を以下に示す。

Excel 上部のリボンから「RExSTM」を選択し（図 10 赤枠 1)、「シート削除」ボタン（図 10 赤枠 2）を選択すると、ダイアログボックスが開き、図 11 が表示される。図 11 で「OK」を選択すると、メイン操作画面以外のシートが削除される。

図 10 「シート削除」ボタン

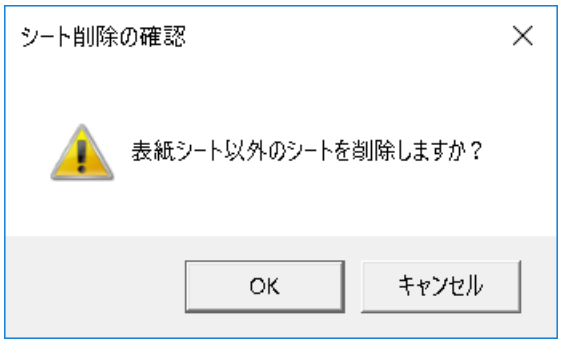

図 11 シート削除の確認

5. 使用上の注意事項

5.1. SmartScreen によるリバース生成の失敗 (Windows8, 10 ユーザ向け)

SmartScreen とは、PC の安全性を保つことを目的として、インターネット上からダウンロードした作成者不明のファイルを実行する際に警告する Windows の機能である。

Excel シート上のボタンを押して条件処理表・状態遷移表のリバース生成を行うとき、Excel がフリーズする、あるいは画面が変化しないといった不具合が発生することがある。この不具合が発生した場合、SmartScreen を一時的に無効にすることにより、不具合を解消できることがある。

以下に、SmartScreen を”無効”にする手順を示す。

（注意）本ツールの実行後、SmartScreen 機能は有効に戻すことを推奨する

(1) コントロールパネルを開く。

(2) コントロールパネルから「セキュリティとメンテナンス」を選択する。

(3) 「セキュリティとメンテナンス」画面左にある「Windows SmartScreen 設定の変更」を選択する。

(4) 図 12 のようなウィンドウが現れたら、下部の「何もしない (Windows SmartScreen を無効にする)」にチェックを入れて、「OK」を選択する（有効に戻す場合は上部にチェックを入れる）。

図 12 SmartScreen のウィンドウ

5.2. 対象ソースコード

以下に本ツールによってサポートされているソースコードの要件を示す。

- GCC によるコンパイルによって、エラーを生じないソースコードであること[警告なし]
- 複雑度 260 以下のソースコードであること[警告なし]
- 最大のネストの深さが 5 以下のソースコードであること[警告あり]
- if と switch が組み合わせられている場合、その構文の最大ネスト数は 3 であること[警告あり]
- 変数と定数が区別されていること[警告なし]
  - ・マクロで定義された定数は、マクロ展開されるため#define を残しておく必要がある
  - ・const 修飾子で宣言された変数は未対応である

なお、[警告なし]と末尾に表記されている項目は、ツールが該当する要件を判定するための機能を持たないことを示している。判定機能は存在しないが、入力ソースコードは本ツールの正常動作保証対象外となる。

また、[警告あり]と末尾に表記されている項目は、本ツールが該当する要件を判定するための機能を持っており、もし該当するソースコードを入力した場合に、処理中のいずれかの段階で警告メッセージが表示される。[警告あり]と末尾に表記されている項目についても、入力ソースコードは本ツールの正常動作保証対象外となる。

5.3. 条件処理表、状態遷移表作成時にエラーが出た場合の対処

4.2 章の条件処理表の作成や 4.3 章の状態遷移表の作成の操作中にエラーが出た場合は、RExSTM/temp, RExSTM/TSV_info, RExSTM/TSV_Result の 3 つのディレクトリ内のファイルをすべて削除し、条件処理表の作成からやり直すことで解決されることがある。

5.4. 条件処理表、状態遷移表作成時に警告が出た場合の対処

4.2 章の条件処理表の作成や 4.3 章の状態遷移表の作成の操作中に図 **13** のような警告画面が出る場合があるが、「実行(R)」を選択すると操作を続けることができる。

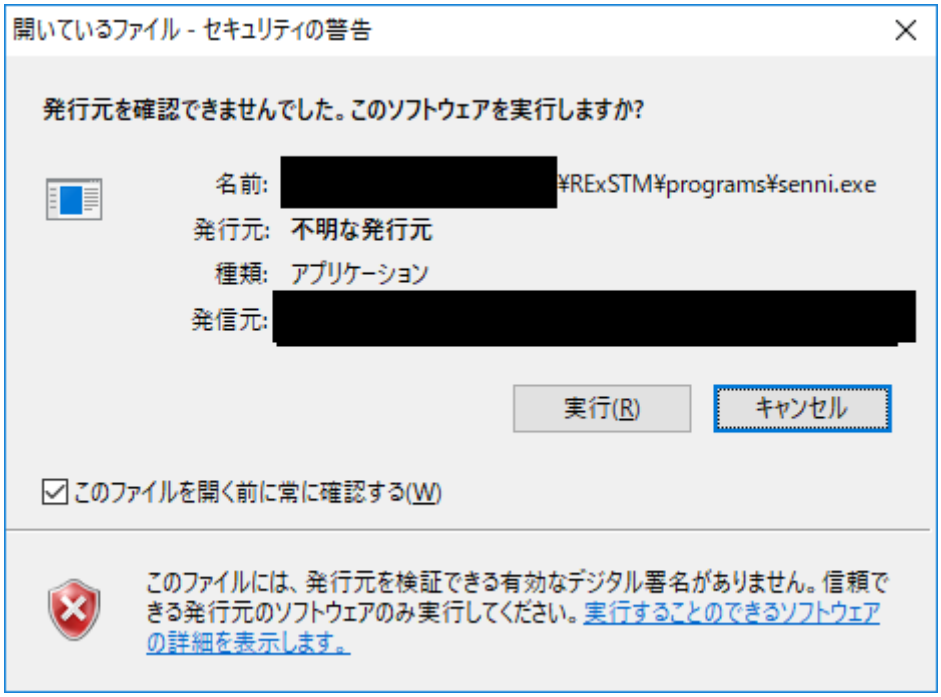

図 13 警告画面

5.5. 状態変数についての制約

状態変数とは if 文、および、switch 文の式で値を比較される単一の変数であり、if 文の場合は以下の(ア)～(オ)を、switch 文の場合は以下の(カ)～(ケ)をすべて満たす変数である。

(ア) if 文の式で値を比較される単一の変数である（式内で比較演算子が使われず、変数のみが指定される場合も含む)。

(イ) (ア)について、式内で状態変数とあわせて使われる比較演算子は、"=="のみとする。

(ウ) 関数内で取りうる状態変数の値の種類は最大 20 である。

(エ) (ア)の構文の内部で更新される変数である。

(オ) 関数の外で宣言されている変数である。

(カ) switch 文の式で値を比較される単一の変数である（式内で演算子が使われず、変数のみが指定される場合も含む)。

(キ) 関数内で取りうる状態変数の値の種類は最大 20 である。

(ク) (カ)の構文の内部で更新される変数である。

(ケ) 関数の外で宣言されている変数である。

6. 不具合情報

本章では既知の不具合について説明する。既知の不具合は、以下の通りである。

6.1. 中括弧の出力について

処理として扱われる条件分岐文（if など)に中括弧が出力されない（図 **14**)。

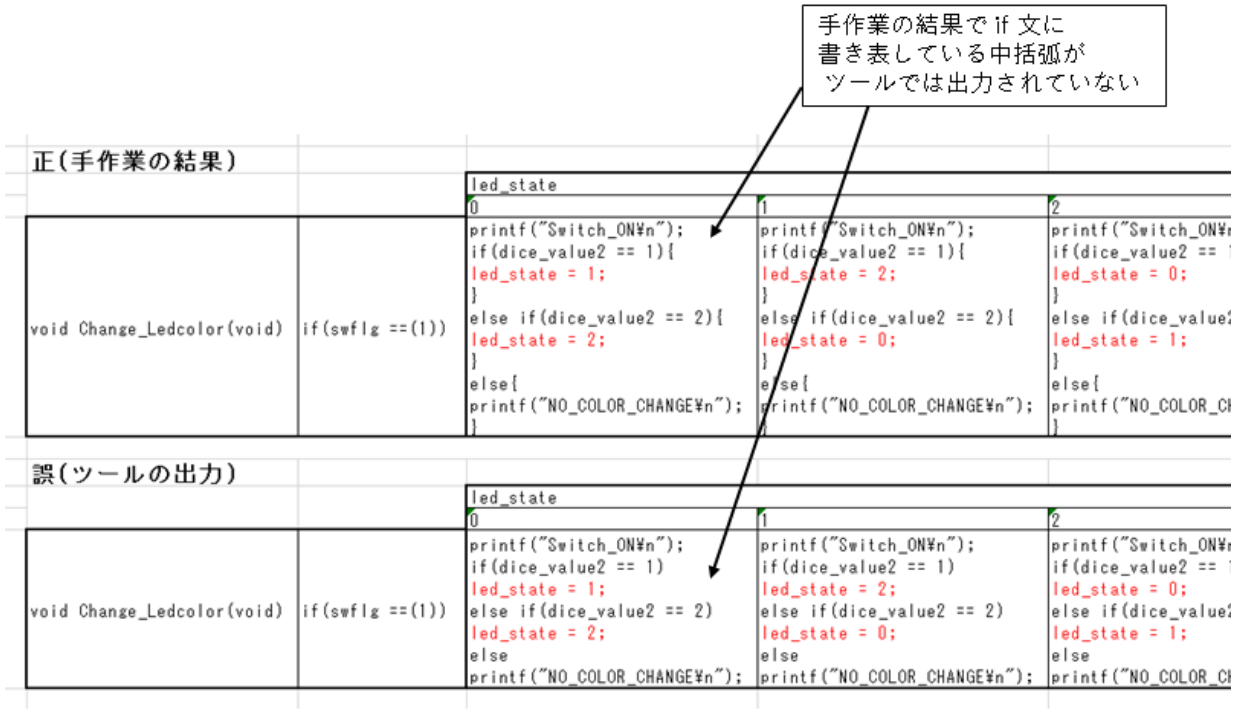

図 14 中括弧が出力されない例

6.2. 状態値に依存しないコードについて

状態値（状態遷移表の列）に依存しないコードが状態遷移表に出力されないことがある（**図 15**)。

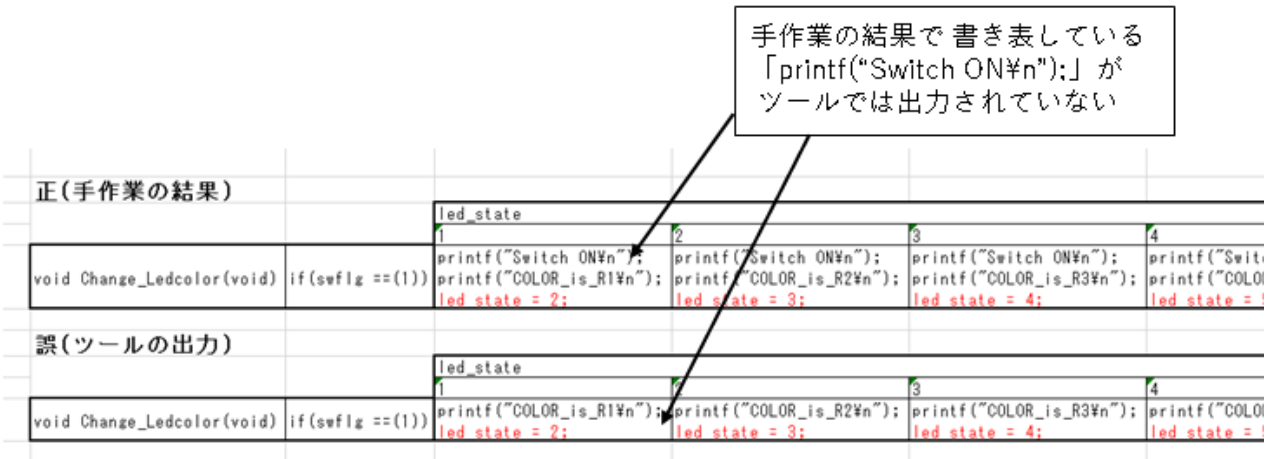

図 15 状態値に依存しないコードが出力されない例