2P-01 動作履歴の時系列パターン抽出によるシステム動作検証*

高橋 理　　石岡卓也†

三菱電機 (株)　産業システム研究所‡

1　はじめに

システムの大規模・多機能化に伴って、システム構造が複雑になり、開発工程における試験／動作検証作業は一層困難になっている。筆者らは、システム内部で実際に行われた処理の順序や内容を正確に検証するために、時系列に沿って記録した動作履歴ファイルを解析するツール TRACS(Task pRofile Analysis tool for Control Systems) [1] を開発している。

本報告では、動作履歴ファイルの中に繰り返し現れる履歴行の組合せを時系列パターンとして抽出することを検討する。これは、時刻項と本文項からなる動作履歴各行を、システム内部で並行して行われる複数の機能処理が単位処理ごとの実行結果を出力したものであると考えた上で、それぞれの機能処理を示すような履歴行の組合せを自動抽出することを目指すものである。また、抽出した時系列パターンを用いて、すべての動作履歴行を分類することにより、障害などの非定型処理を早期に発見することが期待できる。

2　時系列パターンの抽出

本報告では、単位処理の実行状態を示す動作履歴各行を、処理を実行した時刻を示す時刻項と処理の内容を示す本文項の組合せ (T_j, A_j) と表現した上で、システム上で互いに独立して発生する一連の機能処理の流れを示す履歴行の組合せ $\{(T_j, A_j)\}$ を時系列パターンとして定義する。

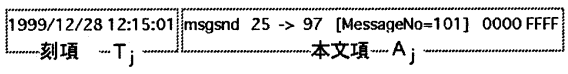

図 1. 動作履歴行の例

一方、複数の処理やデータの流れが混在する並行処理環境下では，一つの処理を構成する履歴行が常に単独で現れるとは限らない。複数の処理が混在する動作履歴から、時系列パターンの数やそれぞれのパターンを構成する履歴行を特定することは大変困難である。

本章では、まず、データ圧縮アルゴリズム LZ78 [2] における辞書作成方法を応用して、一連の処理を構成している可能性が高い履歴行の組合せを時系列パターン候補として登録する。次に、時系列パターン候補の評価によりパターンを最終決定する。

2.1　時系列パターン候補の生成

LZ78 [2] は，入力データを文字列として読み込み，辞書に登録された文字列に一致するかどうかを調べる。そして，最大長の一致文字列の辞書登録番号をその文字列の代わりに出力すると同時に次の一文字を追加した文字列を新たに辞書登録することにより，データ総量の圧縮を行うものである。この辞書登録の作業では，辞書登録済の文字列に一文字ずつ追加して新たな文字列を作成するという手順を経るため，入力データに繰り返し出現する文字列ほど長い辞書登録文字列に含まれ，出現回数の少ない文字列は辞書に登録されにくい。

ここでは、動作履歴の各行を１つの文字に見たて，複数の連続した履歴行を一つの時系列パターン候補として辞書登録する。このとき、辞書の i 番目に登録した n_i 行からなる時系列パターン候補 $(T_j^i, A_j^i)(1 \le j \le n_i)$ と動作履歴の間で，(1) に示す照合を行う。照合が一致した場合には，さらに次の一行を加えた履歴行群を新たなパターン候補として登録する。

$$\forall j(1 \le j \le n_i) \rightarrow \begin{cases} A_j = A_j^i \\ |(T_j - T_1) - (T_j^i - T_1^i)| < \varepsilon_1 \end{cases} \quad (1)$$

2.2　時系列パターン候補の評価

入力動作履歴を末尾まで読み込み，辞書登録 (時系列パターン候補の生成) が完了した後，登録された時系列パターン候補の中から、適切なパターンを選択する。

今、n 行の動作履歴からなる正しい時系列パターン $P_n = \{(T_1, A_1)(T_2, A_2) \cdots (T_n, A_n)\}$ が辞書に登録されているとする。このとき、辞書作成の手順から、既

*System Validation and Verification Method Based on Trace Pattern Mining

†Satoru Takahashi and Takuya Ishioka

‡Industrial Electronics and Systems Laboratory, Mitsubishi Electric Corporation

存の辞書登録履歴行群 (パターン候補) に一致する入力動作履歴が出てくるたびに、次の一行を追加した履歴行群を新たなパターン候補として登録するため、n 行未満のパターン候補 P_i も辞書に登録されていることになる。

$$P_i = \{(T_1, A_1)(T_2, A_2)\cdots(T_i, A_i)\} \quad (i = 1, 2\cdots n-1)$$

さらに、$n+1$ 行以上のパターン候補も辞書登録されている可能性がある。

$$P_j = \{(T_1, A_1)\cdots(T_n, A_n)\cdots(T_j, A_j)\} \quad (j = n+1, n+2\cdots)$$

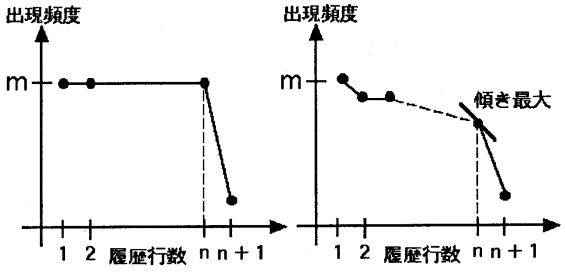

図 2. 履歴行数と出現回数に着目した評価

次に、正しい時系列パターン P_n が m 回だけ出現する動作履歴において、P_n を構成する各履歴行が、他のパターンには含まれず、他のパターンと混在することもなく出力される状況を想定する。このとき、式 (1) に従って入力動作履歴ファイルを再検索し，それぞれの時系列パターン候補 $P_k\,(k = 1, 2\cdots n, n+1\cdots)$ が出現する回数を数えると、図 2 (左) に示すようになる。すなわち、時系列パターン候補の行数が正しい時系列パターンの行数である n 行より少ない場合には、その出現回数は m 回であるが、$n+1$ 行以上の行数をもつパターン候補では、n 行目までの部分 (正しい時系列パターン) と $n+1$ 行目以降 (他の時系列パターンの一部) は互いに独立して発生するとことから、$n+1$ 行目以降の履歴が常に同一の履歴であることはなく、出現回数は少なくなる。したがって、出現回数の変化を用いて、n 行目までの動作履歴を正しい時系列パターンであるとみなすことができる。

一方、実際の動作履歴では、同一種類の履歴行が複数の時系列パターンに含まれる場合があったり、複数の時系列パターンが混在して出力されることがあるため、n 行以下の時系列パターン候補 P_i を構成する履歴行群が連続して出現する回数は、図 2 (右) に示すように、行数 i が増えるにしたがって単調減少する。ただし、正しいパターンを構成する n 行の履歴行間には (1) 式に示す緩やかな依存関係があるのに対して、正しい時系列パターンには含まれない履歴とは独立して発生する。すなわち、n 行未満のパターン候補 P_i から 1 行増えたパターン候補 P_{i+1} における減少度合よりも、正しい時系列パターン P_n から 1 行増えた P_{n+1} における出現回数の減少度合の方がより大きくなると判断できる。そこで、行数／出現回数の傾きが最大になるだけの履歴行をもって時系列パターンであると決定する。

3 時系列パターン抽出事例

列車運行管理システムを構成する 3 種類の計算機から、それぞれ約 6 時間分の動作履歴を取得して時系列パターンの抽出を行った結果を 表 1 に示す。

いずれの場合にも大部分の動作履歴を時系列パターンとして抽出することができた。これらの時系列パターンをシステムの機能や役割と結びつけて解釈することにより、本手法を用いない検証作業と比較して、時系列パターンの抽出率に見合うだけの労力を軽減することが期待できる。今後は、抽出した時系列パターンとシステム機能の関連づけや時系列パターンに照合しない履歴行からの障害推定などを行う所存である。

表 1. 時系列パターンの抽出結果

計算機	A	B	C
実行プロセス数	61	148	18
動作履歴の行数	81852	106182	16033
本文項の種類	399	1119	93
抽出したパターンの種類	14	14	4
識別できなかった履歴行数	13915	27608	521
パターン抽出率	83%	74 %	97%

参考文献

[1] 高橋 理, 石岡卓也 : 動作履歴を用いた監視制御システムの処理解析, 電気学会論文誌, Vol. 120-D, No. 2, 2000 (掲載予定)

[2] Ziv, J. and Lempel, A.: *Compression of individual sequences via variable-rate coding*, IEEE Transactions on Information Theory, Vol. 24, No. 5, pp. 530–536, 1978