2Z-2 探針型デバッグ手法の実現(2) – アルゴリズム

田村 文隆　岡本 渉　植木 克彦　平山 雅之
株式会社東芝 研究開発センター

1 はじめに

ソフトウェアの効率的なデバッグを実現するため、我々は探針型デバッグ手法と呼ぶ手法を提案している[1]。我々は、まず、この手法の対象を逐次プログラムに絞り、ログとして記録するイベントをプログラムの実行位置から得られる情報に限った場合の実装を行った。

本稿では、この手法の基礎となるログ間の類似度や、ログに含まれる特徴値をいかにして定めるかを、簡単な例を用いて概説する。尚、我々の手法の有効性、本稿の計算手法の妥当性の評価に関しては[2]を参照されたい。

2 ログ間の類似度

我々が実装したツールでは、ソースプログラム中にプローブを埋め込み、一意に流れが追えるレベルでプログラムの実行位置をログに記録する。こうして記録したログをユーザに提示し、その中から、エラーの生じた位置と、ログ比較の起点となる位置とを指定させる事により、エラーを含むログと比較対象となるログとを切り出す。

例として図1に示した関数 f(),g() を含むプログラムを考える。ここで、A,B,Cとコメントした部分には何らかの処理が書かれており、位置Aを実行すると異常終了する物とする。同図に示した3種類のログは、比較起点をf()の先頭、エラー位置をAとした時に考えられるパターンである。関数f()が12回呼ばれ、各回に入力値iとして$1,2,\cdots,12$を一度づつ与えた場合、Type1, 2, 3のパターンがそれぞれ、9, 2, 1回現れる。

我々の目的は、上記ログを例にすると、エラーを含むType3に最も特徴の似たログを選び出し、更に、共通する顕著な特徴等の情報を提示する事により、ユーザがエラー原因を特定する際の支援をする事である。これを数値的に行う為、ログの木構造から得られる各種イベントからの寄与の総和として、ログ間の類似度を定義する。

今、ログの集合$S=\{L_0,L_1,\cdots,L_{n-1}\}$を比較対象とする時、各ログ$L_i$ $(0\leq i<n)$から深さ2以上のノードを全て刈って作られるログ$\hat{L}_i$を縮退ログと呼び、これら縮退ログからなる集合を$\hat{S}$と書く。ログ$L_i,L_j\in S$間の類似度を求める際、ログ全体に含まれるイベントの内、我々はまず縮退ログに関するイベントの寄与だけを考え、次に刈り取った部分ログからの寄与を再帰的に考えると言う手法を取る。即ち類似度は$\Delta_S(L_i,L_j)=\Delta_{\hat{S}}(L_i,L_j)+\Delta_{\text{部分ログ}}(L_i,L_j)$と書かれる。

"An Implementation of Probe Debugging Method(2) - Algorithm", Humitaka Tamura, Wataru Okamoto, Katsuhiko Ueki, and Masayuki Hirayama, TOSHIBA corp. R&D Center
e-mail: tamu@ssel.toshiba.co.jp

```
void f() {
  int i;
  iを入力;
  if (i%4 == 0)
  {
    g(i);
  }
}

void g(int j) {
  if (j%3 == 0)
  {
    /* A: Error */
  } else {
    /* B */
    /* C */
  }
}
```

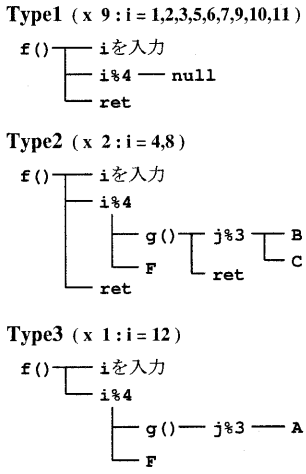

図1: プログラム例とログの木

縮退ログ内のイベントを、互いに対応付けられる物同士グループ化した物をイベント母集団と呼ぶ。集合$\hat{S}$中に考えられるイベント母集団を$E_0,E_1,\cdots,E_{N(\hat{S})-1}$、各$E_k$に含まれるログ$L_i$のイベントを$e_{ki}$と書く。$N(\hat{S})$は$\hat{S}$のみで定まる定数である。あるイベント母集団$E_k$に属す二つのイベント$e_{ki},e_{kj}$が$\Delta_{\hat{S}}(L_i,L_j)$に与える寄与を、$e_{ki},e_{kj}$が等価な時、

$$\delta(e_{ki},e_{kj})=-\log(\Pr(e_{ki})\cdot\Pr(e_{kj}))>=0, \quad (1)$$

又、e_{ki},e_{kj}が非等価な時、

$$\delta(e_{ki},e_{kj})=+\log(\Pr(e_{ki})\cdot\Pr(e_{kj}))<0 \quad (2)$$

と定める。ここで$\Pr(e)$は、イベントeの発生確率で、eに等価なイベントがイベント母集団E_kの中にm個あれば、$\Pr(e)=m/|\hat{S}|$と定義する。式1の等号はイベント母集団に属する全てのイベントが等価な時にのみ成立する。式1,2は、稀にしか起きないイベントは大きな情報量を持つ事を反映している。これらの寄与の総和として、縮退木に関するログ間の類似度を

$$\Delta_{\hat{S}}(L_i,L_j)=\sum_{k=0}^{N(\hat{S})-1}\delta(e_{ki},e_{kj}) \quad (3)$$

と定める。右辺の和を構成する各要素は、対応するイベントの特徴の似通い方を表している。特に、$L_i=L_j$の時、右辺の各要素は、ログL_iの縮退ログに含まれる各イベントの特徴値を表現する。

一方、部分ログの寄与$\Delta_{\text{部分ログ}}(L_i,L_j)$は、ログ$\hat{L}_i$で縮退させた各ノードAから始まる部分ログ$L_i(A)$(但し、

ノードAが存在しない場合は、$L_i(A) = \epsilon$と書く)の寄与の総和である。詳細は後述する。

3 ログの木とイベント

ここでは、イベントをどの様に定めてログ間の類似度を計算するかを示す。但し、ログに含まれるループ構造や再帰構造に対する扱いの説明は紙面の都合で省略する。

ログの集合Sを比較対象とし、ログ$L_i, L_j \in S$の類似度を求める時、縮退木の寄与$\Delta_{\hat{S}}(L_i, L_j)$では、次の二種類のイベントを考慮する:

- 流れイベント – 例: [i を入力 – i%4 – ret]
- 展開イベント – 例: $\langle$i%4$\rangle$, $\overline{\langle \text{i\%4} \rangle}$

流れイベントとは、縮退ログ$\hat{L}_i$の一連の流れを一つのイベントと捉えた物で、各縮退ログ$\hat{L}_i$に対し一つ存在する。集合$\hat{S}$に含まれるログの流れイベントは、皆、同じ一つのイベント母集団に属する。

一方、展開イベントは、ログの集合Sから縮退ログの集合$\hat{S}$を作る際に縮退させた関数や条件分岐等の各ノードAに対し、Aがログ中に存在するか否かを一つのイベントと捉えた物である。ログの集合$\hat{S}$の少なくとも一つのログに縮退させたノードAが含まれるならば、$\hat{S}$の全縮退ログはノードAに関する展開イベント:「Aが存在する」、又は、「Aが存在しない」を持つ。それぞれのイベントを$\langle$A$\rangle$, $\overline{\langle \text{A} \rangle}$と書き、いずれも同じイベント母集団に属すると考える。即ち、縮退ログ$\hat{L}$は、$L(A) \neq \epsilon$ならば、展開イベント$\langle$A$\rangle$を持ち、$L(A) = \epsilon$ならば、展開イベント$\overline{\langle \text{A} \rangle}$を持つ。

次に、ログ$L_i, L_j \in S$の類似度に対する部分ログの寄与$\Delta_{\text{部分ログ}}(L_i, L_j)$について説明する。あるノードAに対し、ログ$L_i, L_j$が、$L_i(A) \neq \epsilon$、かつ、$L_j(A) \neq \epsilon$を満たす場合のみ、ノードAの部分木からの寄与をログL_i, L_jの類似度に加える。ノードAの部分木からの寄与は、Aから始まる部分木の集合$S(A) = \{L(A) \mid L \in S, L(A) \neq \epsilon\}$を新たな比較対象として再帰的に計算した部分ログ$L_i(A), L_j(A)$の類似度として定義される。従って、部分ログからの寄与は、

$$\Delta_{\text{部分ログ}}(L_i, L_j) = \sum_{\text{A s.t.} L_i(A), L_j(A) \neq \epsilon} \Delta_{S(A)}(L_i(A), L_j(A)). \quad (4)$$

と書かれる。

一般に、あるノードAに対し、$L_i(A) \neq \epsilon, L_j(A) \neq \epsilon$の時は、式4の右辺は、部分ログを再帰的に計算するにつれ、違いが累積されて大きな負の値を取る事になる。これに対し、縮退木に関する展開イベントはこの負の値を緩和する働きをする。

4 例

ここでは、図1の例でログ間の類似度を計算して、前章までの内容を具体的に示す。

エラーを含むType3のログ i = 12 とType1のログの一つ i = 1 との間の類似度は、

$$\begin{aligned}
\Delta_S(L_{12}, L_1) &= +\log(\frac{1}{12} \cdot \frac{11}{12}) - \log(\frac{12}{12} \cdot \frac{12}{12}) \quad (5) \\
&\quad + \Delta_{S\langle \text{i\%4} \rangle}(L_{12}(\langle \text{i\%4} \rangle), L_1(\langle \text{i\%4} \rangle)) \\
\text{第3項} &= +\log(\frac{3}{12} \cdot \frac{9}{12}) + \log(\frac{3}{12} \cdot \frac{9}{12}) \\
\therefore \Delta_S(L_{12}, L_1) &\simeq (-2.57) + 0 + ((-1.67) + (-1.67)) \\
&\simeq -5.92 \quad \text{(自然対数)} \quad (6)
\end{aligned}$$

式5で、第1,2項はそれぞれ、流れイベント、展開イベントからの寄与、第3項は部分ログからの寄与である。同様に、Type3のログとType2のログとの間の類似度、Type3のログとそれ自身との類似度を求めると、

$$\begin{aligned}
\Delta_S(L_{12}, L_4) &\simeq -2.57 + 0 + (+2.77 + 2.77 \\
&\quad + (-1.50 + 0 + (-1.50))) \\
&\simeq -0.03 \quad (7) \\
\Delta_S(L_{12}, L_{12}) &\simeq +4.97 + 0 + (+2.77 + 2.77 \\
&\quad + (+2.20 + 2.20 + 0)) \\
&\simeq +14.9 \quad (8)
\end{aligned}$$

となる。以上の結果より、エラーのログに最も似たログは、エラーのログ自身を除けば、Type2の二つのログである事が分かる。更に、式6,7の各項を比較すれば、第3項、即ち図1のi%4に相当する部分で、Type2のログはType1のログより大きな値を持ち、これが両者の類似度の違いに最も貢献している事が分かる。又、エラーを含むログ自身の特徴を表す式8の各項を比較すると、やはり第3項のi%4が最も大きく貢献している。以上より、i%4の部分に最もエラーの原因が含まれている可能性が高い事が分かった。そして、本例の場合、これはバグの原因と一致している。

5 まとめ

本稿では、我々が提案した探針型デバッグ手法をツール化した際に用いた類似度の計算方法を概説した。これはログの木構造を再帰的に辿り、イベントの発生確率を元にした情報量を累積して行くと言う方法である。ここで提案した計算方法に関しては、文献[2]の評価実験によってその妥当性を確認済みである。但し、本稿では触れなかったが、実際のログを解析する場合には、ループや再帰呼出しに対する考慮が不可欠となる。我々の評価実験では、ループまで考慮した類似度を用いている。ループや再帰呼出しの扱い方は続報する。

参考文献

[1] 植木 克彦他, "探針型デバッグ手法の提案", 信学技報 Vol.100, No.186, pp.1-8 (2000-7)

[2] 岡本 渉他, "探針型デバッグ手法の実現 (1) – 概要と評価", 62回情処学会全国大会 2Z-1 (2001-03)