大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

トップエスイー ソフトウェア開発実践演習

ソフトウェアの障害調査に使用可能な検証モデル自動生成

株式会社 富士通コンピュータテクノロジーズ　浅川 春奈　asakawa.Haruna@jp.fujitsu.com

開発における問題点

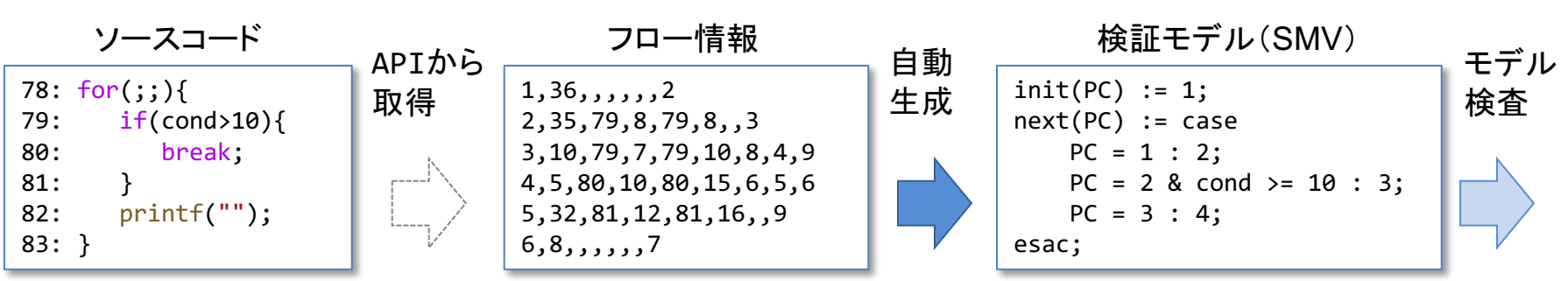
ソフトウェアシステムにける再現率が低い不具合の原因特定のため、静的解析ツールが効果的な例がある。
しかし一方で、静的解析ツールでは検出が困難な種類の不具合もあり、原因特定に繋がらない例がある。

手法・ツールの適用による解決

静的解析ツールが特に苦手とする並行システムの検証のため、モデル検査を障害調査に適用する。調査が難航するケースにおいては、仕様書が存在しない例も多いため、ソースコードから検証モデルへの自動生成する手法の検討を行った。

取り組み内容：SMV自動生成ツール

ソースコードからモデル解析言語のSMVを自動生成するツールを開発した。
モデル検査においては、モデルの「状態」を全パターン検証するため、プログラムの「状態」を表す要素を検証モデルに変換する。「1. プログラムの実行位置」「2. 変数の値変化」に着目して変換ツールを設計・実装した。また、「1. プログラムの実行位置」については、ソフトウェア構造解析ツールのAPIから取得できるフロー情報を利用して短期間で実現した。



評価

①自動変換の対象範囲
「1.プログラムの実行位置」の変換については、分岐・ループ・関数ジャンプの対応より、十分な範囲の自動変換が実現できた。
②実行可能なSMVの出力
静的解析ツールでは問題検出ができないサンプルコードにおいて、自動生成した検証モデルを使用して問題を検出できた。

無限ループの検証例

```
1: void test(){
2:     for(int i=0;i<5;i=i+1){
3:         function_call();
4:         i = i - 1;
5:     }
6: } // PC = 18
```

検証式：AF(PC=18)

今後の取り組み

- 自動変換対象の言語・構文を拡大する
- 静的解析ツールとモデル検査を併用し、機械的に問題検出が可能な領域を拡大していく。モデル検査の効率化のため、静的解析ツールの「チェッカー」に相当する、モデル検証パターンを拡充する。