Software Fault Localization and Analysis using Data Mining and Visualization

Prof. Dr. Stefan Leue

DFG Colloquium
Konstanz
26 June 2008

Contributors: E. Stegantova, G. di Fatta (group of M. Berthold), S. Leue
Cooperation within GK: M. Berthold

1. Fault Localization in Software Testing

I. Function Call Tree recording and abstraction
• fct corresponds to context of fault

II. Frequent Subtree Mining

Observations:
• Siemens test suite
• 7 correct prgms, plus 129 variants with injected faults
• 7-21 functions per pgm, 173-565 loc per function
• multiple thousand tests

Function ranking:
• "naive" ranking
• FP ranking
• occurrence of function in suspicious subtrees

G. Di Fatta, S. Leue, E. Stegantova: Discriminative pattern mining in software fault detection. Proc. of ACM SOQUA’06

Contributors: N. Ali, S. Leue
Cooperation within GK: M. Berthold

2. Fault Localization in Model Checking

Observations:
• model possesses 4109 deadlock traces (as determined by model checker SPIN)

Questions:
• how many deadlocks?
• which are the causal factors (=errors)

Idea:
• use of data mining algorithms to detect frequent patterns in deadlocked traces that are not frequent in non-deadlocked traces
• in particular, sequence mining algorithms

Challenges:
• "good" traces?
• approximation: deadlock-free lassos
• sequence mining and scalability
• non-conclusiveness of patterns (gaps)
• faulty patterns are not known
• pattern representation (reg. exp.?)

Approach:
• sequence / episode mining based on presumed patterns
• development of algorithms that elicit patterns

One pattern causing a deadlock:
offhook_onhook_offhook_onhook

3. Visualization for Software Debugging

State Space Graph (Stochastic Model) with Counterexample

Traces Deviation Visualization

Contributors: H. Aljazzar, N. Ali, S. Leue

Cooperation within GK:
• Powerwall (Keim)
• Interaction with gigapixel displays (Reiterer)
• graph drawing (Brandes)


Challenges:
• "good" traces?
• approximation: deadlock-free lassos
• sequence mining and scalability
• non-conclusiveness of patterns (gaps)
• faulty patterns are not known
• pattern representation (reg. exp.?)

Approach:
• sequence / episode mining based on presumed patterns
• development of algorithms that elicit patterns

Contributors: H. Aljazzar, N. Ali, S. Leue

Cooperation within GK:
• Powerwall (Keim)
• Interaction with gigapixel displays (Reiterer)
• graph drawing (Brandes)


Challenges:
• "good" traces?
• approximation: deadlock-free lassos
• sequence mining and scalability
• non-conclusiveness of patterns (gaps)
• faulty patterns are not known
• pattern representation (reg. exp.?)

Approach:
• sequence / episode mining based on presumed patterns
• development of algorithms that elicit patterns

Contributors: H. Aljazzar, N. Ali, S. Leue

Cooperation within GK:
• Powerwall (Keim)
• Interaction with gigapixel displays (Reiterer)
• graph drawing (Brandes)


Challenges:
• "good" traces?
• approximation: deadlock-free lassos
• sequence mining and scalability
• non-conclusiveness of patterns (gaps)
• faulty patterns are not known
• pattern representation (reg. exp.?)

Approach:
• sequence / episode mining based on presumed patterns
• development of algorithms that elicit patterns

Contributors: H. Aljazzar, N. Ali, S. Leue

Cooperation within GK:
• Powerwall (Keim)
• Interaction with gigapixel displays (Reiterer)
• graph drawing (Brandes)