Mining Message Flows from SoC Execution Traces

Md Rubel Ahmed, Hao Zheng   U of South Florida

Motivation

Formal Specifications are useful in
1. Testing
2. Verification
3. Maintenance
4. Validation
5. Understanding the system
6. …

But also difficult and costly to write, therefore often missing or incomplete.
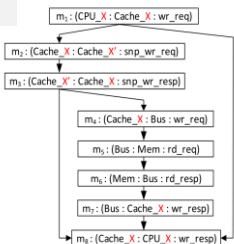
Goal

Fig. 1: CPU downstream write Flow

FlowMiner aims to extract $\langle m_1, m_8 \rangle$, $\langle m_1, m_2, m_3, m_8 \rangle$, $\langle m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8 \rangle$
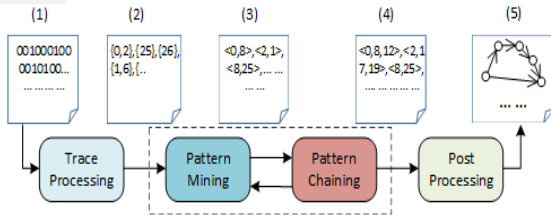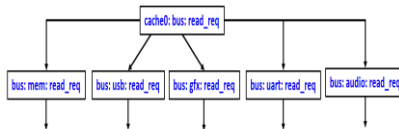
Method

Fig. 2: Flow mining steps

Contributions
1. A sequential pattern mining framework for highly concurrent SoC traces
2. A set of inference techniques
3. A set of optimization methods for reducing search space
4. Comparative evaluation with state-of-the-art specification mining tools

Scope

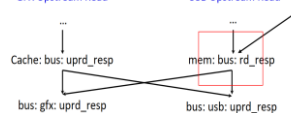*FlowMiner* can successfully extract flows that share either a common suffix, or a common prefix

Results

Traces	Tool	#Pattern mined	Precision	Recall
Single-non interleaved	FM [1] [2]	40 30 26	100% 100% 100%	18.7% 6.25% 0.00%
Single-interleaved	FM [1] [2]	109 2 0	46% 100% 0%	18.7% 0.00% 0.00%
Mult-interleaved	FM	114	43%	18.7%

Future works

References:
[1] Jinlin Yang, David Evans, Deepali Bhardwaj, Thirumalesh Bhat, and Manuvir Das. Perracotta: Mining temporal api rules from imperfect traces. In Proceedings of the 28th International Conference on Software Engineering, ICSE '06, pages 282–291, New York, NY, USA, 2006. ACM.
[2] Lingyi Liu and Shobha Vasudevan. Automatic generation of system level assertions from transaction level models. Journal of Electronic Testing, 29(5):669–684, Oct 2013.