





AGORA: Automated Generation of Test Oracles for REST APIs

Team members: Wadood Alam, Bhagyashree Basudkar, Shashank
Betawar, Souvik Das



Project Design

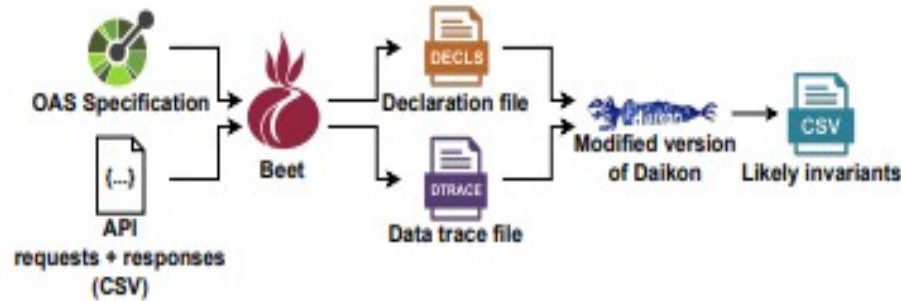


Figure 1: Workflow of AGORA.

1. OAS file for API
2. Generate API Req+Res
3. Use Beet to get trace and declaration file
4. Use modified Daikon to generate likely invariants
5. Invariant analysis to find errors

Research Questions 1

What is the effectiveness of AGORA in generating test oracles for all operations/endpoints of each API, and does it align with the reported precision trends by the author's usage of limited set of endpoints for testing and evaluation?

Why worth answering:

- We feel that there is no conclusive evidence to prove the comprehensiveness(in terms of precision) when test for all endpoints of an API.
 - Practical use case

Approach

- Conduct comprehensive evaluation of AGORA's effectiveness in generating test oracles for all operations/endpoints of each API.
- Determine if reported trends by authors hold for all operations of each API.
- Utilize 5 of the 7 APIs used by authors for reference comparison of precision trends.

Evaluation

- Using 5 out of the 7 APIs used by the authors for comparison of results with authors' findings to assess validity of precision trends across diverse APIs.
- These 5 APIs have specification file available publicly.
- Comparison with default version of Daikon on the same dataset for benchmarking.
- Manual classification of true-positive or false-positive invariants for precision assessment.

Research Questions 2

What is the impact of changes made to the endpoint parameters or internal code on the precision of the results obtained from testing the precision of developed endpoints?

Why worth answering:

- It is a common occurrence for code changes to API
 - Important to understand the effectiveness of AGORA in this real-world use case

Approach

- Change endpoint parameters or internal code
- Check generated invariants
- Evaluate precision

Evaluation

- Evaluation of RQ2 involves focusing on a single endpoint for assessment.
- Code modifications made to alter endpoint, potentially leading to changes in detected invariants by AGORA.
- Comparison of invariants detected between unmodified and modified code versions. Both datasets used as input to AGORA for modified endpoint to identify any discrepancies in generated invariants.



QUESTIONS?