AGORA: Automated Generation of Test Oracles for REST APIs

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

What are Test Oracles?

Test Oracle is a mechanism, different from the program itself, that can be used to **test the accuracy of a program's output for test cases**. Conceptually, we can consider testing a process in which test cases are given for testing and the program under test. The output of the two then compares to determine whether the program behaves correctly for test cases.

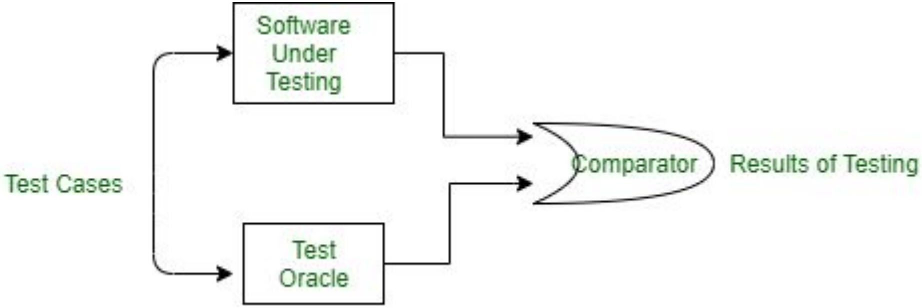


Figure - Testing and Test Oracles

Project Design

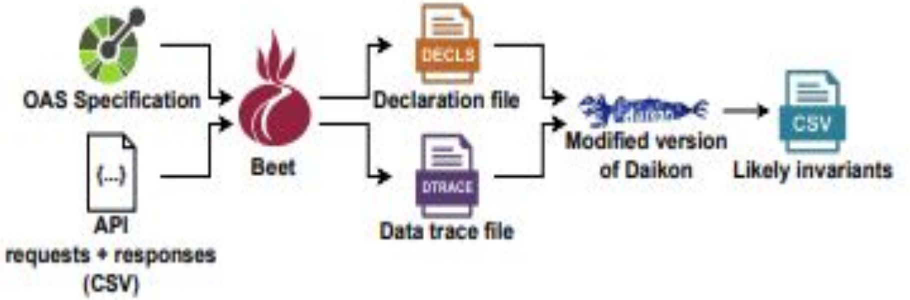


Figure 1: Workflow of AGORA.

1. OAS file for API
2. Generate API Req+Res using RESTest
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

RQ1: How effective is AGORA in generating test oracles?

RQ2: What is the impact of the size of the input dataset on the precision of AGORA?

RQ3: How effective are the generated test oracles in detecting failures?

RQ4: Do these existing results’ trend of AGORA stand true for different APIs?

Data Used in Evaluation

Data Sources:

- Varied test inputs for each experiment.
- 11 operations from 7 different APIs.
- Diverse set of API requests and responses for input diversity.

API Specifications:

- Leveraged OpenAPI Specification (OAS) of APIs under test.
- Utilized publicly available API specifications whenever possible.
- Manually generated specifications for OMDb and Yelp due to unavailability.

Extended data:

- 2 more APIs with at least 4 operations
- Variable API requests

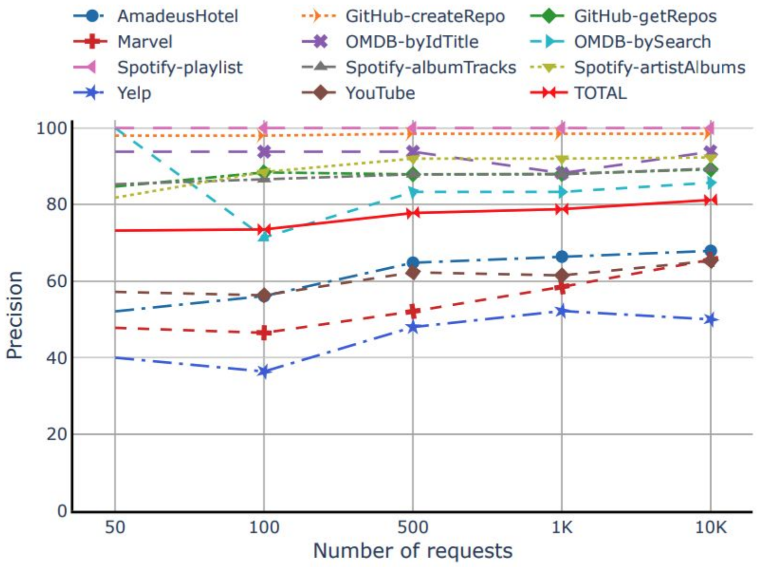
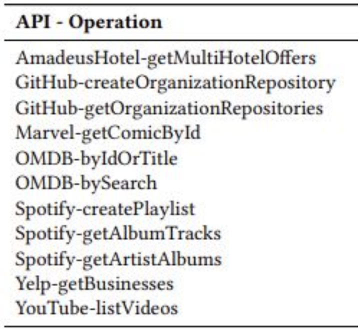


Figure 3: Evolution of the precision of AGORA.



API - Operation
AmadeusHotel-getMultiHotelOffers
GitHub-createOrganizationRepository
GitHub-getOrganizationRepositories
Marvel-getComicById
OMDB-byIdOrTitle
OMDB-bySearch
Spotify-createPlaylist
Spotify-getAlbumTracks
Spotify-getArtistAlbums
Yelp-getBusinesses
YouTube-listVideos

Evaluation - Effectiveness and Comparison

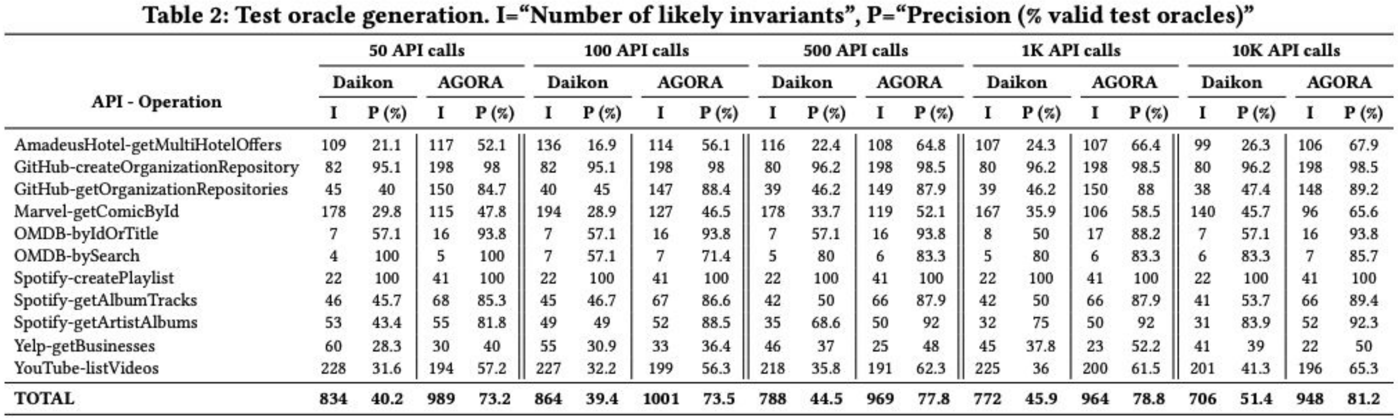


Table 2: Test oracle generation. I="Number of likely invariants", P="Precision (% valid test oracles)"

API - Operation	50 API calls				100 API calls				500 API calls				1K API calls				10K API calls			
	Daikon		AGORA		Daikon		AGORA		Daikon		AGORA		Daikon		AGORA		Daikon		AGORA	
	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)	I	P (%)
AmadeusHotel-getMultiHotelOffers	109	21.1	117	52.1	136	16.9	114	56.1	116	22.4	108	64.8	107	24.3	107	66.4	99	26.3	106	67.9
GitHub-createOrganizationRepository	82	95.1	198	98	82	95.1	198	98	80	96.2	198	98.5	80	96.2	198	98.5	80	96.2	198	98.5
GitHub-getOrganizationRepositories	45	40	150	84.7	40	45	147	88.4	39	46.2	149	87.9	39	46.2	150	88	38	47.4	148	89.2
Marvel-getComicById	178	29.8	115	47.8	194	28.9	127	46.5	178	33.7	119	52.1	167	35.9	106	58.5	140	45.7	96	65.6
OMDB-byIdOrTitle	7	57.1	16	93.8	7	57.1	16	93.8	7	57.1	16	93.8	8	50	17	88.2	7	57.1	16	93.8
OMDB-bySearch	4	100	5	100	7	57.1	7	71.4	5	80	6	83.3	5	80	6	83.3	6	83.3	7	85.7
Spotify-createPlaylist	22	100	41	100	22	100	41	100	22	100	41	100	22	100	41	100	22	100	41	100
Spotify-getAlbumTracks	46	45.7	68	85.3	45	46.7	67	86.6	42	50	66	87.9	42	50	66	87.9	41	53.7	66	89.4
Spotify-getArtistAlbums	53	43.4	55	81.8	49	49	52	88.5	35	68.6	50	92	32	75	50	92	31	83.9	52	92.3
Yelp-getBusinesses	60	28.3	30	40	55	30.9	33	36.4	46	37	25	48	45	37.8	23	52.2	41	39	22	50
YouTube-listVideos	228	31.6	194	57.2	227	32.2	199	56.3	218	35.8	191	62.3	225	36	200	61.5	201	41.3	196	65.3
TOTAL	**834**	**40.2**	**989**	**73.2**	**864**	**39.4**	**1001**	**73.5**	**788**	**44.5**	**969**	**77.8**	**772**	**45.9**	**964**	**78.8**	**706**	**51.4**	**948**	**81.2**

- AGORA compared to default Daikon configuration.
  - Achieved total precision of 81.2% with 10K API requests, an improvement of 29.8%
    - Manually: Classifying False/True positives.
- Test with 2 more APIs
  - Verifying if this trend persists or not

Project Timeline

Week 3-4: Project Kickoff and Research

- Conduct a detailed study of the paper
- Select APIs for testing.

Week 5-6: Data Collection and Preparation

- Acquire or generate API specifications
- Generate API requests and responses.
- Set up a environment(RESTest)

Week 7: Implementation of Beet/Daikon

- Run beet on local machine
- Generate data trace and declaration file
- Conduct Daikon(default) work

Week 8: AGORA Implementation

- Run AGORA on local machine
- Conduct AGORA work and generate invariants

Week 9: Results and Presentation Preparation

- Prepare results from AGORA and Daikon
- Creation of presentation

Responsibilities

All team members collaboratively contribute to every phase of the project.

Bhagyashree : Project Coordinator

- Oversee project progress.
- Coordinate communication within the team.
- Find 2 APIs and its OAS

Souvik: RESTest

- Run RESTest
- Generate API Req/Res
- Run default Daikon(new APIs)

Wadood: Beet

- Run Beet
- Generate Declaration and Data trace files

Shashank: AGORA

- Run modified version of daikon
- Generate likely invariants