Improving Test Case Generation for RESTful APIs through Seeded Sampling

Chiel de Vries

Date: June 25th 2020
Course: CSE3000
Supervisors: A. Panichella
M.J.G. Olsthoorn
P. Pawelczak

1. Context

Background:
- Automated testing of RESTful APIs with EvoMaster
- White-box Testing
- Evolutionary algorithm
- Existing sampling methods:
  - Random sampling
  - Smart Sampling

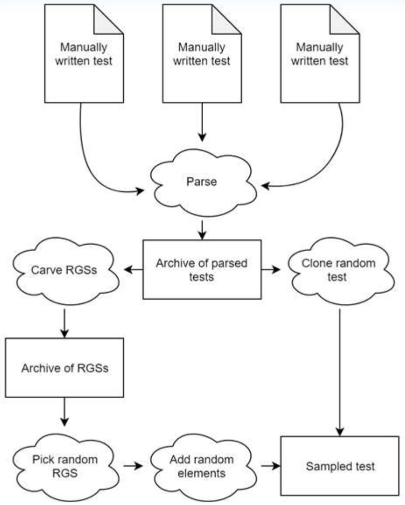
Terms:
RESTful API: Webservice using HTTP requests and responses to handle resources
Seeding: Using previous knowledge
Sampling: Initialization of tests

Goal:
Improve the coverage achieved by test suites generated with EvoMaster by exploiting manually-written test cases.

2. Seeded Sampling

The elements of seeded sampling:
- Parser: parses test cases to internal representation
- Sampler: Clones or carves from parsed test cases

Terms:
Cloning: Copy a parsed test
Carving: Extract RGS and add random elements
RGS: Resource Generating Sequence, a sequence of POST/PUT requests


Manually written test
Manually written test
Manually written test
Parse
Carve RGSs
Archive of parsed tests
Clone random test
Archive of RGSs
Pick random RGS
Add random elements
Sampled test

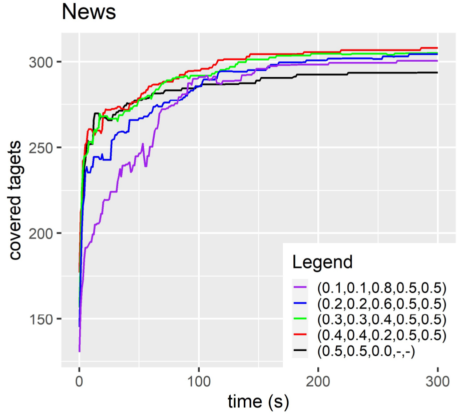
3. Evaluation

Research Question:
To what extent can seeded sampling improve coverage compared to the current combination of sampling techniques used by EvoMaster?

Evaluation settings:
- Tested on 2 APIs
- 7 Parameter sets used
- 10 repetitions per set
- Runs of 5 minutes

Parameter Set:
A set of probabilities for sampling
P_{random}: Random sampling
P_{smart}: Smart sampling
P_{seeded}: Seeded sampling
P_{clone}: Cloning
P_{carve}: Carving
P_{clone} and P_{carve} are the probabilities given seeded sampling is chosen

Result:
Better performance when P_{seeded} is low (< 0.4). However improvements are small. Coverage is improved by no more than 2 percent points


News

4. Limitations

Internal:
- Simple parser
- Few repetitions

External:
- Few Apis tested