Sujet de PSTL : Benchmarking de solutions optimistes pour génération de données test à partir de JSON-Schema

Encadrants : Mohamed-Amine Baazizi (mohamed-amine.baazizi@lip6.fr),

Lyes Attouche <lyes.attouche@dauphine.psl.eu>

Objectifs

JSON-Schema [1] est un langage d'assertions assez expressif permettant de décrire la structure des données JSON. Son usage tend à se généraliser (description des APIs, vérification de pipelines d'apprentissage, métadonnées, etc) et il devient crucial de pouvoir générer des instances de tests à partir de schémas en entrée. Différentes implémentations permettant une telle génération automatique mais ne garantissent pas toujours un résultat correct dans le sens où elles peuvent générer des instances non conformes au schéma en entrée. Parmi ces implémentations, certaines adoptent une approche optimiste qui consiste à générer une instance en examinant les fragments des schémas et en les combinant dans l'espoir que l'instance obtenue soit valide et peuvent même avoir recours à un validateur externe dans l'optique de réparer les instances qui seraient non conformes. Ces approches ont pour point commun leur efficacité puisqu'elles ne souffrent pas de la complexité du problème de génération qui aussi difficile que la satisfiabilité et pour lequel il a été démontré une complexité exponentielle [7].

Le but de ce projet est d'étudier les limitations théoriques et pratiques de ces approches en ayant recours à la fois à de la rétro-ingénierie et à l'analyse expérimentale. Cette analyse portera, principalement, sur trois librairies open-source déjà identifiée et choisies pour leur prise en compte de la quasi-totalité des opérateurs du langage de schéma [1,2,3] mais pourra s'étendre à d'autres librairies open-source qu'on jugera pertinentes. Le projet s'inscrit dans le cadre d'un projet en cours ayant déjà permis de développer une approche correcte et complete mais qui peut s'avérer couteuse [5, 6] dans certains cas.

• Tâches à réaliser

1. Etude de l'état de l'art pour compréhende JSON SCHEMA et la génération d'instances à partir d'un schéma
2. Prise en main et retro-ingineire des 3 librairies afin de décrire leur fonctionnement
3. Analyse expérimentale en utilisant des schémas réels et synthetiques

• Prérequis

- Maîtrise de Java, Javascript et C# langages utilisés par les trois librairies identifiées
- Sens analytique et attrait pour la formalisation de problemes avec finalité pratique

• Références

- [1] https://json-schema.org
- [2] Jsongenerator. https://github.com/jimblackler/jsongenerator
- [3] Json schema faker. https://github.com/json-schema-faker/json-schema-fakeryes
- [4] JSON everything. https://github.com/gregsdennis/json-everything
- [5] Lyes Attouche, Mohamed Amine Baazizi, Dario Colazzo, Francesco Falleni, Giorgio Ghelli, Cristiano Landi, Carlo Sartiani, Stefanie Scherzinger: A Tool for JSON Schema Witness Generation. EDBT 2021: 694-697
- [6] Mohamed Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, Stefanie Scherzinger: Not Elimination and Witness Generation for JSON Schema. CoRR abs/2104.14828 (2021)
- [7] Pierre Bourhis, Juan L. Reutter, Fernando Suárez, Domagoj Vrgoc . JSON: Data model, Query languages and Schema specification. PODS'17