UNIVERSITÉ DE FRANCHE-COMTÉ

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

école doctorale **sciences pour l'ingénieur et microtechniques** SPIM

Titre de la thèse/Thesis title : Approche neuro-symbolique pour générer un référentiel de test

Laboratoire d'accueil / Host Laboratory : FEMTO-ST / Département DISC

Spécialité du doctorat préparé/Speciality : Informatique

Mots-clefs / Keywords : Test, Intelligence Artificiel, Contrainte

Descriptif détaillé de la thèse / Job description

La complexité des systèmes fait qu'il est nécessaire de mettre en place une assurance qualité (QA). Cette dernière s'intègre dans le cycle de vie du système (ou de l'application) entre l'analyse du cahier des charges à sa maintenance après déploiement. La QA s'appuie principalement sur l'utilisation des tests (unitaire, fonctionnel, intégration, sécurité…).

La réalité du terrain est que la conception, la rédaction et l'exécution de tests est majoritairement manuelle. Aujourd'hui des environnements dédiés aux tests apparaissent, permettant une partie d'automatisation de leur exécution et une aide à la rédaction. Pour la conception, la démarche proposée est basée sur des modèles. Cependant, elle nécessite un investissement en temps et en ressources s'appuyant sur des compétences rares. Ainsi, le travail est souvent réalisé sur des sous-ensembles fonctionnels ou sécuritaires justifiant l'investissement.

Pourtant l'intérêt et la valeur ajoutée sont indéniables (rationalisation, couverture, détection d'ambiguïté…). Cette thèse vise à apporter des éléments pour accompagner l'ingénieur validation dans la conception et la génération d'un patrimoine de test. Il doit ainsi pouvoir se référer aux exigences issues du cahier des charges et sur les différents artéfacts à sa disposition (traces utilisateurs, tests, anomalies…).

Pour arriver à cela, il doit pouvoir intégrer l'ensemble de ces éléments. Il doit pouvoir établir un référentiel qui lui permet d'avoir une vision de la couverture de son système. Comme illustré dans la Figure 1, à partir d'un cahier des charges (1), contenant des éléments permettant d'assurer la traçabilité, une première IA (LLM) préalablement entraînée sera capable de générer des tests "bruts" composés de mots clés abstraits [1] (keyword-based testing) décrivant des séquences d'actions à réaliser sur le système (2). Une seconde IA, symbolique cette fois, basée sur la résolution de contraintes, sera chargée d'instancier les cas de tests. Il s'agira à cette étape d'inférer automatiquement des appels d'opérations et des valeurs de paramètres (d'entrées et de sorties) pour celles-ci de façon à pouvoir exécuter les tests sur le système avec les valeurs attendues (3).

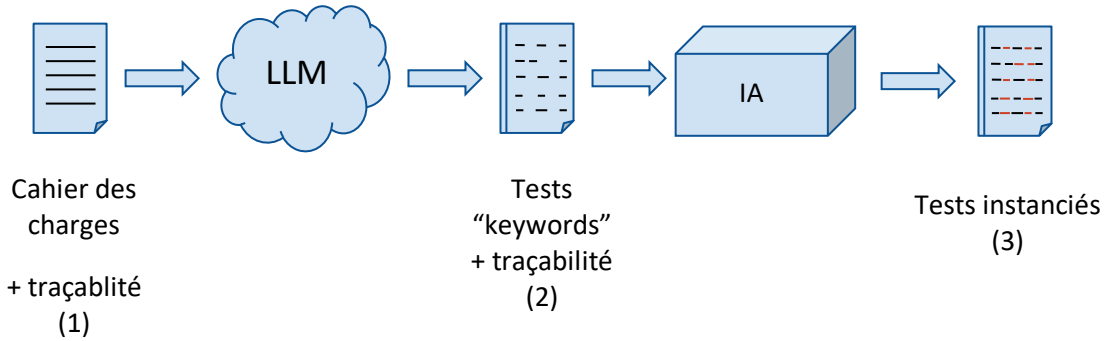

Figure 1 - Processus de génération des tests

Cette thèse se déroule au sein du département d'informatique des systèmes complexes de l'institut Femto-ST. Elle s'inscrit dans une thématique plus globale de recherche sur l'IA pour la QA. Elle vise à apporter des réponses théoriques, méthodologiques et outillées pour accompagner l'ingénieur validation dans la conception et la gestion de son patrimoine de test. Il doit ainsi pouvoir se référer aux exigences issues du cahier des charges pour établir un ou plusieurs modèles. Ces derniers se serviront de ce référentiel pour permettre d'avoir une vision de la couverture des tests de son système et aider aux pilotages de son référentiel de test. Celui-ci peut ainsi évoluer et suivre le rythme des livraisons, grâce à cette démarche modulaire.

Objectif de la thèse et problématique scientifique

Sur la base de la création de ce référentiel, l'enjeu de ce travail est de pouvoir établir les séquences de tests. Une approche basée sur l'utilisation de modèles pose souvent le problème de l'explosion combinatoire de l'espace de recherche (ensemble des actions activables à un point donné ou ensembles des valeurs des paramètres). Pour ce faire, nous voulons mettre en place une architecture permettant d'utiliser les approches d'IA génératives pour aider à établir les séquences [2] et ainsi réduire l'espace de recherche. Le risque avec ce type d'approche est la consistance du choix. Généralement, celui-ci ne pourra être établi qu'une fois les données d'entrée fournies. De plus, elle ne permet pas d'obtenir les valeurs attendues, car ces approches ne sont pas faites pour faire des calculs. Pour éviter cela et permettre d'obtenir les valeurs attendues, nous couplerons l'approche avec l'IA Symbolique. Notre savoir-faire dans le domaine des solveurs de contraintes et leurs usages dans le test [3], [4], nous permet d'envisager une collaboration fine entre les deux mondes comme proposé dans [5], [6] permettant ainsi d'établir les séquences de tests.

L'originalité de ce sujet est de proposer une approche mélangeant ce qui pourrait être le meilleur des deux mondes de l'IA : symbolique et générative dans le domaine du test. Cette approche commence à être identifiée dans la littérature sous le nom de neuro-symbolique [7].

En résumé, le travail de thèse devra répondre aux questions de recherche suivantes :

- Peut-on utiliser les IA génératives pour synthétiser des cahiers des charges dans un but d'obtenir des tests :
    - Est-ce que les IA génératives sont suffisamment fiables dans leurs analyses pour permettre d'obtenir des séquences candidates pour établir les tests ?
    - Peut-on utiliser l'IA symbolique sur la base des éléments produits par l'IA générative pour permettre d'effectuer les calculs nécessaires pour établir ou valider les séquences et les données de tests ?
- Peut-on optimiser un référentiel de test tout en donnant des garanties de couvertures :
    - Comment détecter l'inclusion de traces de tests, en cas de tests manuels harmonisation des mots clés pour optimiser le patrimoine ?
    - Comment établir une vigie pour évaluer la couverture du référentiel de test ?
    - Peut-on compléter le patrimoine de test vis-à-vis de la couverture ?
- Peut-on construire un environnement logiciel permettant le traitement de bout en bout de la gestion du patrimoine de test :
    - Quel type d'architecture logicielle pour permettre l'intégration des différents composants ?
    - Quels algorithmes et points de contrôles définir pour permettre au testeur de piloter le patrimoine ?

Méthodologie pour la thèse et déroulement du travail

Le travail de cette thèse peut se découper de la façon suivante :

- T0 - T0+8 : Analyse des LLMs open-source pré-entrainés (tels que Mistral 7B ou Zephyr 7B) et ingénieur des prompts [8], [9], [10] et fine-tuning [11] pour étudier l'extraction de l'information des cahiers des charges. Nous pourrons nous baser sur des benchmarks standard, tels que celui de HuggingFace [12]. Ensuite, nous utiliserons des ensembles de données issus de partenaires. Nous sommes actuellement en discussion avec le service informatique et données du grand Besançon ainsi que des acteurs du monde du test (Agilitest, Smartesting, XQual…).
- T0+6 - T0+14 : Intégration des résultats entre les IA générative [13] et symbolique pour obtenir des séquences de tests logiques.
- T0+14 - T0+24 : Optimisation des interactions entre les IA.
- T0+24 - T0+36 : Usinage (*refactoring* en anglais) des tests existants et/ou des traces utilisateurs pour obtenir ou établir la vigie qui servira de référence pour établir les métriques et donc la confiance dans le référentiel de tests.

À cela s'ajoutent :

- des étapes de rédaction pour la valorisation des résultats de recherche dans les conférences et journaux des domaines du test et de l'IA.
- la volonté de confronter la démarche à différents systèmes réels pour affiner les algorithmes et la pertinence des tests obtenus.

Références bibliographiques / Bibliography

[1] R. Hametner, D. Winkler, et A. Zoitl, « Agile testing concepts based on keyword-driven testing for industrial automation systems », in IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, oct. 2012, p. 3727-3732. doi: 10.1109/IECON.2012.6389298.

[2] M. Utting, B. Legeard, F. Dadeau, F. Tamagnan, et F. Bouquet, « Identifying and Generating Missing Tests using Machine Learning on Execution Traces », in 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), Oxford, United Kingdom: IEEE, août 2020, p. 83-90. doi: 10.1109/AITEST49225.2020.00020.

[3] F. Bouquet, F. Dadeau, B. Legeard, et M. Utting, « Symbolic Animation of JML Specifications », in FM 2005: Formal Methods, vol. 3582, J. Fitzgerald, I. J. Hayes, et A. Tarlecki, Éd., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 75-90. doi: 10.1007/11526841_7.

[4] F. Ambert et al., « BZ-TT: A Tool-Set for Test Generation from Z and B using Constraint Logic Programming », in Proceedings of the CONCUR'02 Workshop on Formal Approaches to Testing of Software (FATES'02), R. Hierons et T. Jerron, Éd., Brnö, Czech Republic: INRIA Report, août 2002, p. 105-120.

[5] P. Giadikiaroglou, M. Lymperaiou, G. Filandrianos, et G. Stamou, « Puzzle Solving using Reasoning of Large Language Models: A Survey ». arXiv, 17 février 2024. doi: 10.48550/arXiv.2402.11291.

[6] J. Chu-Carroll et al., « Beyond LLMs: Advancing the Landscape of Complex Reasoning ». arXiv, 12 février 2024. doi: 10.48550/arXiv.2402.08064.

[7] Z. Wan et al., « Towards Cognitive AI Systems: a Survey and Prospective on Neuro-Symbolic AI ». arXiv, 2 janvier 2024. doi: 10.48550/arXiv.2401.01040.

[8] B. Lester, R. Al-Rfou, et N. Constant, « The Power of Scale for Parameter-Efficient Prompt Tuning », in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, p. 3045-3059. doi: 10.18653/v1/2021.emnlp-main.243.

[9] DeepSeek-AI et al., « DeepSeek LLM: Scaling Open-Source Language Models with Longtermism ». arXiv, 5 janvier 2024. doi: 10.48550/arXiv.2401.02954.

[10] E. J. Hu et al., « LoRA: Low-Rank Adaptation of Large Language Models ». arXiv, 16 octobre 2021. doi: 10.48550/arXiv.2106.09685.

[11] K. Tian, E. Mitchell, H. Yao, C. D. Manning, et C. Finn, « Fine-tuning Language Models for Factuality ». arXiv, 14 novembre 2023. doi: 10.48550/arXiv.2311.08401.

[12] « Open LLM Leaderboard - a Hugging Face Space by HuggingFaceH4 ». Consulté le: 26 mars 2024. [En ligne]. Disponible sur: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

[13] W. Junjie, H. Yuchao, C. Chunyang, L. Zhe, W. Song, et W. Qing, « Software Testing with Large Language Models: Survey, Landscape, and Vision », [En ligne]. Disponible sur: https://arxiv.org/pdf/2307.07221.pdf

Profil demandé / Applicant profile

- Etudiant avec Bac+5 en informatique (master ou ingénieur)
- Une expérience préalable en recherche, que ce soit par le biais de stages, de projets de recherche ou de publications, est souvent appréciée.
- Avoir des connaissances dans l'un des types d'IA en lien avec le sujet.
-

Preferred selection criteria:
- Résultats académiques et classement
- Motivation pour la recherche
- Capacité à communiquer de manière claire et concise, à la fois à l'écrit et à l'oral

Personal characteristics:
- Curiosité intellectuelle
- Rigueur scientifique
- Savoir et pouvoir collaborer
- Persévérance et créativité

Financement : MESRI Etablissement

Dossier à envoyer pour le **27 mai 2024**
Début du contrat : 1er octobre 2024
Salaire mensuel brut : 2 100€

Direction de la thèse:/ **Thesis Supervisor**

BOUQUET Fabrice, fabrice.bouquet@univ-fcomte.fr

Encadrement de la thèse : **co-directeur(s) et co-encadrant(s)**

Frédéric Dadeau, co-encadrant
Dorine Tabary, co-encadrant

Applicants are invited to submit their application to the PhD supervisors.
Application must contain the following documents:
- CV
- Cover letter
- At least 1 reference letter

UNIVERSITÉ DE FRANCHE-COMTÉ

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

école doctorale **sciences pour l'ingénieur et microtechniques** SPIM

Thesis title: Neuro-symbolic approach to generating a test repository

Host Laboratory: FEMTO-ST / DISC Department

Specialty: Computer Science

Keywords: Test, Artificial Intelligence, Constraints

Job description:

The complexity of systems makes it necessary to implement quality assurance (QA). QA is an integral part of the system's (or application's) lifecycle, from analysis of specifications to post-deployment maintenance. QA relies mainly on the use of tests (unit, functional, integration, security...).

The reality in the field is that the design, writing and execution of tests is largely manual. Today, test-dedicated environments are appearing, enabling part of the test execution to be automated, and providing writing assistance. For design, the proposed approach is based on models. However, it requires an investment in time and resources, drawing on rare skills. As a result, work is often carried out on functional or safety-related sub-assemblies that justify the investment.

Yet the benefits and added value are undeniable (rationalization, coverage, ambiguity detection...). The aim of this thesis is to provide information to help validation engineers design and generate test assets. He must be able to refer to the requirements derived from the specifications and the various artifacts at his disposal (user traces, tests, anomalies, etc.).

To achieve this, he needs to be able to integrate all these elements. It must be able to establish a repository that gives it a vision of its system's coverage. As illustrated in Figure 1, given a set of specifications (1), containing elements to ensure traceability, a first AI (LLM) trained beforehand will be able to generate "raw" tests composed of abstract keywords [1] (keyword-based testing) describing sequences of actions to be carried out on the system (2). A second, symbolic AI, based on constraint resolution, will instantiate the test cases. At this stage, it will automatically infer operation calls and parameter values (inputs and outputs) for them, to be able to run the tests on the system with the expected values (3).

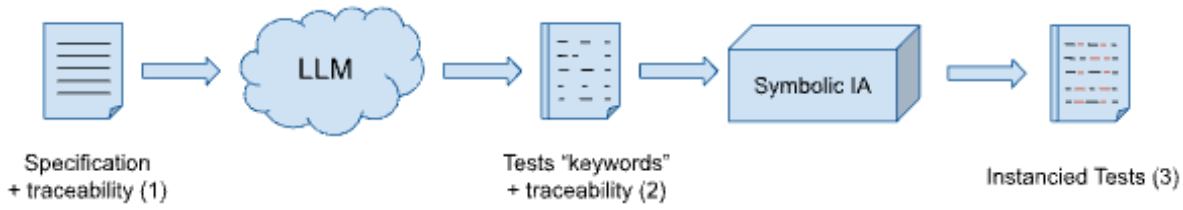

Figure 1 – Test generation process

This thesis is being carried out at the Femto-ST Institute's Complex Systems Computer Science Department (DISC). It is part of a more global research theme on AI for QA. It aims to provide theoretical, methodological, and tool-based solutions to support validation engineers in the design and management of their test assets. He must be able to refer to the requirements derived from the specifications to establish one or more models. These models will be used to provide an overview of the system's test coverage, and to help manage the test repository. Thanks to this modular approach, the repository can evolve and keep pace with deliveries.

Thesis Objective and Scientific Problem

Based on the creation of this repository, the aim of this work is to establish test sequences. A model-based approach often poses the problem of the combinatorial explosion of the search space (set of actions that can be activated at a given point or set of parameter values). To this end, we want to set up an architecture that enables generative AI approaches to be used to help establish sequences [2] and thus reduce the search space. The risk with this type of approach is the consistency of the choice. Generally, it can only be established once the input data has been provided. What's more, the expected values cannot be obtained, as these approaches are not designed for calculations. To avoid this, and to obtain the expected values, we will couple the approach with Symbolic AI. Our know-how in the field of constraint solvers and their use in testing [3], [4], allows us to envisage a fine collaboration between the two worlds as proposed in [5], [6], thus enabling us to establish test sequences.

The originality of this subject is to propose an approach mixing what could be the best of both worlds of AI: symbolic and generative in the testing domain. This approach is beginning to be identified in the literature as neuro-symbolic [7].

In summary, the thesis work will have to answer the following research questions:

- Can generative AIs be used to synthesize test specifications?
    - Are generative AIs sufficiently reliable in their analyses to obtain candidate sequences for establishing tests?
    - Can symbolic AI be used based on elements produced by generative AI to perform the calculations needed to establish or validate test sequences and data?
- Can we optimize a test repository while guaranteeing coverage?
    - How can we detect the inclusion of test traces, in the case of manual testing? How can we harmonize key words to optimize assets?
    - How can we set up a watch to assess test repository coverage?
    - Can test assets be completed regarding coverage?
- Is it possible to build a software environment enabling end-to-end management of test assets?
    - What type of software architecture is needed to integrate the various components?
    - What algorithms and control points should be defined to enable the tester to manage the assets?

Thesis methodology and workflow

The work of this thesis can be scheduled as follows:

- T0 - T0+8: Analysis of pre-trained open-source LLMs (such as Mistral 7B or Zephyr 7B) and prompt engineer [8], [9], [10] and fine-tuning [11] to study the extraction of information from specifications. We can base our work on standard benchmarks, such as HuggingFace [12]. We will then use data sets from partners. We are currently in discussion with the IT and data department of Greater Besançon, as well as with players in the testing world (Agilitest, Smartesting, XQual...).
- T0+6 - T0+14: Integration of results between generative [13] and symbolic AI to obtain logical test sequences.
- T0+14 - T0+24: Optimization of interactions between AIs.
- T0+24 - T0+36: Refactoring of existing tests and/or user traces to obtain or establish the watch that will serve as a reference for establishing metrics and thus confidence in the test repository.

In addition:

- editorial stages to promote research results in conferences and journals in the fields of testing and AI.
- the desire to test the approach on different real-life systems to refine the algorithms and the relevance of the tests obtained.

Bibliography

[1] R. Hametner, D. Winkler, et A. Zoitl, « Agile testing concepts based on keyword-driven testing for industrial automation systems », in IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, oct. 2012, p. 3727-3732. doi: 10.1109/IECON.2012.6389298.

[2] M. Utting, B. Legeard, F. Dadeau, F. Tamagnan, et F. Bouquet, « Identifying and Generating Missing Tests using Machine Learning on Execution Traces », in 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), Oxford, United Kingdom: IEEE, août 2020, p. 83-90. doi: 10.1109/AITEST49225.2020.00020.

[3] F. Bouquet, F. Dadeau, B. Legeard, et M. Utting, « Symbolic Animation of JML Specifications », in FM 2005: Formal Methods, vol. 3582, J. Fitzgerald, I. J. Hayes, et A. Tarlecki, Éd., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 75-90. doi: 10.1007/11526841_7.

[4] F. Ambert et al., « BZ-TT: A Tool-Set for Test Generation from Z and B using Constraint Logic Programming », in Proceedings of the CONCUR'02 Workshop on Formal Approaches to Testing of Software (FATES'02), R. Hierons et T. Jerron, Éd., Brnö, Czech Republic: INRIA Report, août 2002, p. 105-120.

[5] P. Giadikiaroglou, M. Lymperaiou, G. Filandrianos, et G. Stamou, « Puzzle Solving using Reasoning of Large Language Models: A Survey ». arXiv, 17 février 2024. doi: 10.48550/arXiv.2402.11291.

[6] J. Chu-Carroll et al., « Beyond LLMs: Advancing the Landscape of Complex Reasoning ». arXiv, 12 février 2024. doi: 10.48550/arXiv.2402.08064.

[7] Z. Wan et al., « Towards Cognitive AI Systems: a Survey and Prospective on Neuro-Symbolic AI ». arXiv, 2 janvier 2024. doi: 10.48550/arXiv.2401.01040.

[8] B. Lester, R. Al-Rfou, et N. Constant, « The Power of Scale for Parameter-Efficient Prompt Tuning », in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, p. 3045-3059. doi: 10.18653/v1/2021.emnlp-main.243.

[9] DeepSeek-AI et al., « DeepSeek LLM: Scaling Open-Source Language Models with Longtermism ». arXiv, 5 janvier 2024. doi: 10.48550/arXiv.2401.02954.

[10] E. J. Hu et al., « LoRA: Low-Rank Adaptation of Large Language Models ». arXiv, 16 octobre 2021. doi: 10.48550/arXiv.2106.09685.

[11] K. Tian, E. Mitchell, H. Yao, C. D. Manning, et C. Finn, « Fine-tuning Language Models for Factuality ». arXiv, 14 novembre 2023. doi: 10.48550/arXiv.2311.08401.

[12] « Open LLM Leaderboard - a Hugging Face Space by HuggingFaceH4 ». Consulté le: 26 mars 2024. [En ligne]. Disponible sur: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

[13] W. Junjie, H. Yuchao, C. Chunyang, L. Zhe, W. Song, et W. Qing, « Software Testing with Large Language Models: Survey, Landscape, and Vision », [En ligne]. Disponible sur: https://arxiv.org/pdf/2307.07221.pdf

Applicant profile

- Student with 5 years' higher education in computer science (master's degree or engineer)
- Previous research experience, whether through internships, research projects or publications, is often appreciated.
- Knowledge of one of the types of AI related to the subject.

Preferred selection criteria:

- Academic results and ranking
- Motivation for research

- Ability to communicate clearly and concisely, both orally and in writing.

Personal characteristics:

- Intellectual curiosity
- Scientific rigor
- Ability to collaborate
- Perseverance and creativity

Financement : MESRI Etablissement

Dossier à envoyer pour le 27 mai 2024
Début du contrat : 1er Octobre 2024
Salaire mensuel brut : 2100€

Thesis Supervisor

BOUQUET Fabrice, fabrice.bouquet@univ-fcomte.fr

Encadrement de la thèse : **co-directeur(s) et co-encadrant(s)**

- Frédéric Dadeau, co-encadrant
- Dorine Tabary, co-encadrant

Applicants are invited to submit their application to the PhD supervisors.
Application must contain the following documents:

- CV
- Cover letter
- At least 1 reference letter