



Recrutement ING1

Équipe IMAGE

Équipe Automates & Vérification

Décembre 2020





Le LRDE

Généricité & Performance



<https://www.lrde.epita.fr>

Laboratoire de Recherche et Développement de l'EPITA

2 équipes, 16 permanents, 1 chercheur associé, 1 post-doc, 7 doctorants, 7 ING3, 6 ING2, ?? ING1, des stagiaires...

Vous !

...beaucoup de monde est passé par le LRDE !

Comment postuler ?

Postuler ici avant le dimanche 20 décembre 2020 à 23h42

<https://www.lrde.epita.fr/wiki/Home>

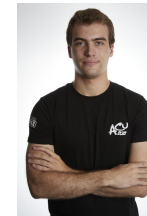


QCM sur Moodle le 23 décembre 2020 (vous recevez un email).

Si vous êtes retenu, vous serez contacté par votre encadrant qui vous donnera quelque chose à faire pendant le mois de janvier pour finaliser (ou pas) le recrutement.

Équipe Automates & Vérification

L'équipe Automates & Vérification



L'équipe Automates & vérification

Sujets principaux

- Manipulation d' ω -automates, de formules de logique temporelle.
- Applications au model checking ou à la synthèse de contrôleur.

Spot (<https://spot.lrde.epita.fr/>)



- bibliothèque C++17, avec des bindings Python, des outils en ligne de commande, et aussi des fonctionnalités accessibles sur le web
- des utilisateurs dans plusieurs équipes de recherche à travers le monde

Sujets proposés par l'équipe Automates & vérification



Intégration des interpolants dans les SAT solveurs parallèles

Encadrants : Etienne Renault et Anissa Kheireddine

Résumé : Les SAT solveurs résolvent le problème de savoir si il existe une interprétation des variables qui satisfait une formule booléenne. Quand ils sont combinés avec des interpolants, ils constituent une solution efficace au problème du model-checking (borné). L'idée est de pouvoir les utiliser dans le cadre de notre solveur parallèle (<https://gitlab.lrde.epita.fr/ludovic/painless>). Cela implique :

1. D'implémenter et de comparer les différentes techniques de traduction d'interpolation existantes
2. De les intégrer dans le SAT solveur parallèle

Compétences attendues/développées : C++, parallélisme, SAT-solveurs, model-checking

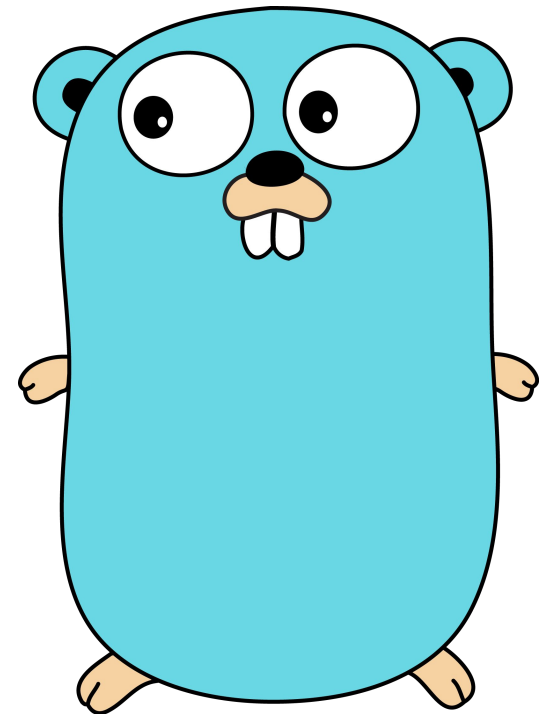


Amélioration des Black-boxes dans Go2Pins

Encadrant : Etienne Renault

Résumé : Go2Pins (<https://gitlab.lrde.epita.fr/spot/go2pins>) est un outil permettant de vérifier formellement des programmes écrits en Go. Pour faire face au problème d'explosion combinatoire, Go2Pins utilise la notion de black-boxes : une technique d'abstraction permettant d'agréger plusieurs calculs en une unique opération. Cette technique est pour le moment limitée aux fonctions pures (sans effets de bord). L'idée est d'adapter ce mécanisme pour qu'il supporte les fonctions avec effets de bord.

Compétences attendues/développées : Golang, techniques de compilation





Apprentissage actif de langages

Encadrant : Adrien Pommellet

Résumé : Comment deviner quelque chose en utilisant le minimum d'information ? L'apprentissage actif est une méthode où un élève doit deviner un modèle (en pratique, un automate) avec pour seule source d'information un professeur qui peut valider ou invalider une suggestion. Il s'agit alors de voir quelles sont les bonnes questions à poser pour trouver une réponse juste au plus vite.

Compétences attendues/développées : C++, bonne connaissance de THLR, automates en tout genre



Synthèse de lois de contrôle pour des robots industriels

Encadrant : Philipp Schlehuber

Résumé : Le but du projet est de synthétiser des lois de contrôle pour des robots industriels à partir des spécifications LTL. La synthèse des contrôleurs se formalise typiquement comme un jeu du contrôleur contre l'environnement. L'objectif est d'intégrer des solveurs pour jeux spécifiques dans Spot et d'étudier l'existant sur la synthèse qualitative et bornée. Une phase finale du projet consistera à mettre en oeuvre ces lois.

Compétences attendues/développées : Synthèse, Logique Temporelle Linéaire (LTL)



Type reflection in Python

Encadrants : Jim Newton

Résumé : The Python language has primitive type reflection capability. We want to use the existing API to build a simple type system atop Python which expresses union, intersection, and complement types. Types in this simple type system should have ability to compute habitation, as well as disjointness, and equality relations. Similar programs are available for reference in Common Lisp, Clojure, and Scala.

Compétences attendues/développées : Python



Comparison of Clojure/spec vs Clojure/genus

Encadrants : Jim Newton

Résumé : The Clojure programming language (a modern lisp dialect) has an extensive library for specifying the “shape” of data in terms of predicates. Clojure/spec makes lofty claims of bridging the gap between statically-typed and dynamic languages. However, the library notably is missing a robust reflective layer. Clojure/genus is a simple type system based on type calculus which provides introspection, but its scope is much smaller than Clojure/spec. This research project involves characterizing these differences and experimenting with ways of using them together.

Compétences attendues/développées : Clojure, functional languages

Équipe IMAGE

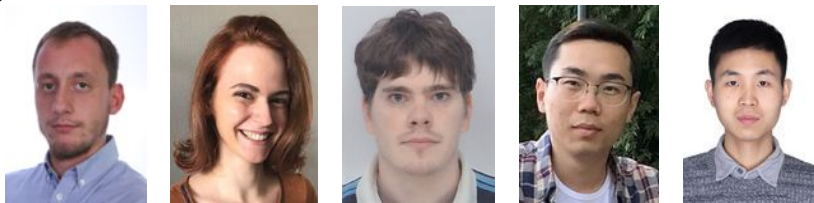
—

L'équipe IMAGE en bref : les gens

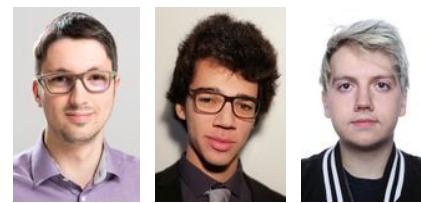


Enseignants-chercheurs

Doctorants



ING3



ING2



L'équipe IMAGE en bref : les axes de travail

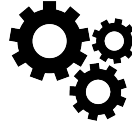


Axe logiciel

Bibliothèque de traitement d'images
générique et performante

<http://gitlab.lrde.epita.fr/olena/pylene>

Langage C++ avancé



Axe applicatif

Dématérialisation de documents
Segmentation en imagerie médicale
avec/sans deep learning
Analyse d'image satellitaires



Axe méthodologique

Morphologie mathématique
Géométrie discrète
Algorithmique

Sujets proposés par l'équipe IMAGE

Implémentation efficace des représentations hiérarchiques morphologiques

Encadrants : Edwin Carlinet, Baptiste Esteban

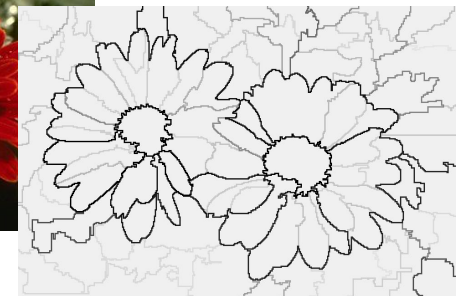
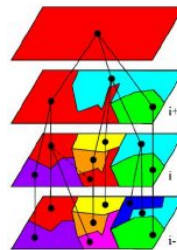
Résumé : Pylene est une bibliothèque de traitement d'images générique et performante qui met à disposition des blocs de traitement morphologiques. Parmi ces traitements, certains reposent sur des représentations hiérarchiques de l'image. L'objectif est de compléter l'offre avec de nouvelles représentations hiérarchiques et leur opérateurs.

Compétences attendues/développées :

C++/Python skills

Devops (Gitlab/CI, CMake...)

Intérêt pour le TI/Morph. Math.



Détection d'objets quasi-linéaires

Encadrants : Joseph Chazalon, Edwin Carlinet, Elodie Puybareau

Résumé : Implémenter une vieille méthode de **détection de « segments »** potentiellement coupés, courbés, bruités, etc. en **C++** pour l'intégrer dans la bibliothèque *open source* de l'équipe. **Version GPU** en option. Cette méthode est rapide, puissante et permet de supprimer un segment tout en conservant les pixels appartenant à d'autres segments. Projet avec un objectif **très clair** et **publication potentielle**.

Compétences attendues/développées :

Cuda/C++ skills

Intérêt pour la performance

Devops (Gitlab/CI, CMake...)

Intérêt pour le TI

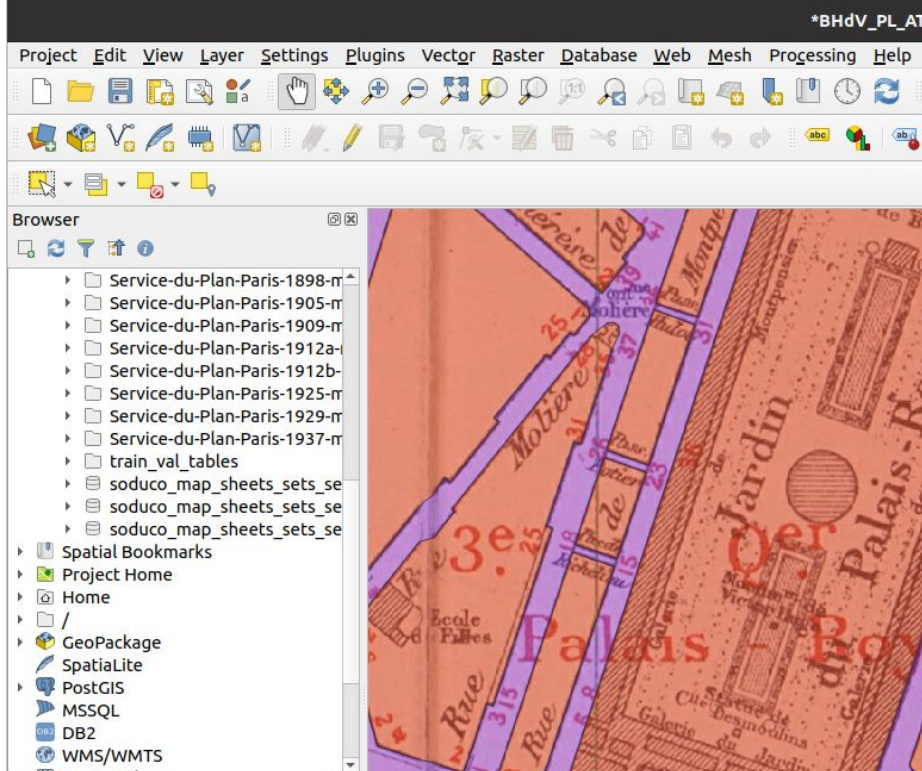


Feedback: $\frac{1}{1 + G(s)H(s)}$ $\frac{Y(s)}{X(s)}$

Résumé : Créer des démonstrateurs pour les techniques d'analyse de plan anciens en les intégrant dans des plugins QGIS (logiciel cartographique **open source**). Dans le cadre d'un projet **avec l'IGN**. Code **Python** et **C++**, architecture **client/serveur**, mise en production d'un système basé **deep learning**. Projet **orienté développement** avec **publication potentielle**.

<https://github.com/soduco/paper-dgmm2021>

Compétences nécessaires : Python, C++



Comparaison de fonctions de coût (deep learning)

Encadrant : Joseph Chazalon

Résumé : Pour entraîner un réseau de neurones, il faut disposer d'une fonction qui donne un coût aux erreurs qu'il commet. Pour une tâche donnée, plusieurs fonctions peuvent être utiles, mais elles guideront l'apprentissage différemment. Il s'agit ici de **poursuivre des travaux déjà démarrés (ING1)** pour **implémenter des fonctions de coût** et **réaliser un comparatif**. Un projet pour se mettre sérieusement au **deep learning** avec **publication potentielle**. **Python, Keras/Tensorflow, PyTorch.**

Compétences nécessaires : Python, Mathématiques

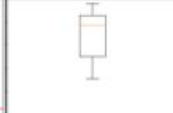
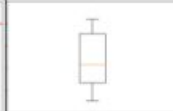
CNN Model	Loss Function 1	Loss Function 2	Metric	Final Performance
Loss Function 1				
Loss Function 2				

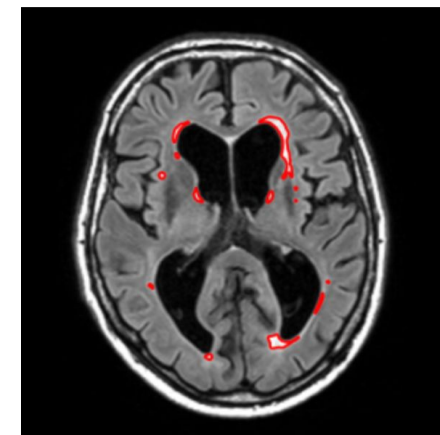
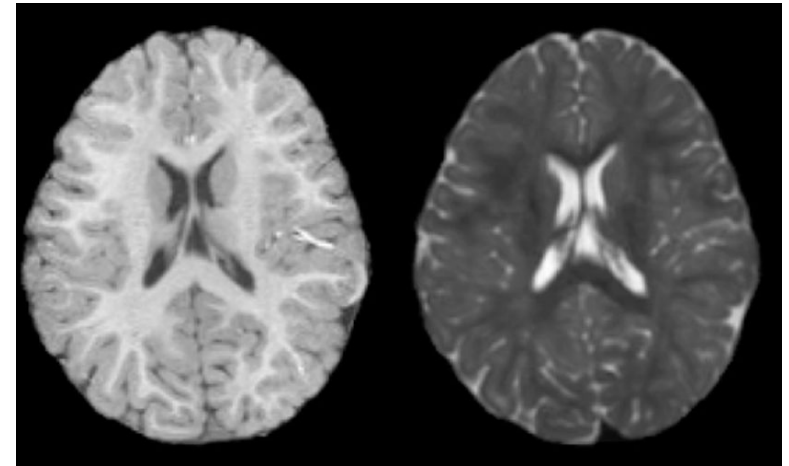
Figure 2.7: Shape of a benchmark table for a CNN model

SOS de codes en détresse

Encadrante : Elodie Puybareau

Résumé : Implémenter des anciennes versions de codes python (2.7, anciennes bibliothèques, etc.) dans un python récent afin de pouvoir les partager (Github). Les applications sont dans les domaines médicaux, avec ou sans deep learning. L'idée est de reproduire les résultats, de documenter les algos. Le but final : partager les codes avec la communauté internationale, et les améliorer !

2 principales applications : imagerie cérébrale et microscopie

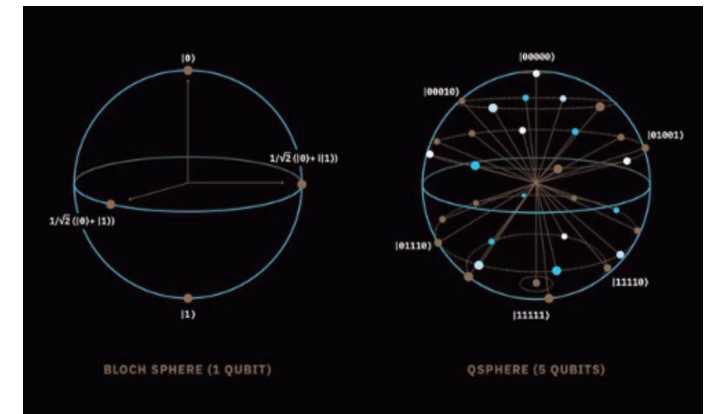


Informatique quantique et Intelligence Artificielle

Encadrant : Nicolas Boutry

Résumé : l'informatique quantique a montré des capacités incroyables de réduction de complexité ces dernières années grâce à la superposition d'états et à l'intrication. Pour cette raison, ce projet a pour but de voir comment le Quantum Computing permettra de réduire la complexité d'un sujet très gourmand en calcul, l'Intelligence Artificielle. Des algorithmes devront être élaborés dans ce but et pour cette raison, de bonnes capacités de raisonnement en algorithmique sont requises.

<https://www.youtube.com/watch?v=Lbndu5EIWvI>



Intéressé·e ?

—

Et maintenant ?

Contactez les encadrants ou des projets qui vous intéressent par email.

<https://www.lrde.epita.fr/wiki/People>

Vous devez postuler avant le dimanche 20 décembre 2020 à 23h42

<https://www.lrde.epita.fr/wiki/Home>

