

Projet de fin d'étude : Développement d'une IA symbolique pour l'avionique embarquée, implémentée sur nouvelles architectures de calcul (GPU, TPU, FPGA)

Mots-clés : IA symbolique, raisonneur, logique descriptive, calcul parallèle, avionique embarquée, GPU/TPU/FPGA

Accueil : LAAS-CNRS, équipe TRUSTworthy systems

Encadrement : Philippe Leleux

Youssef Amari

Hélène Waeselynck

Dates : Février à août 2025

Contexte

Dans le domaine de l'avionique, les systèmes embarqués doivent gérer une quantité croissante de données critiques, notamment celles issues des NOTAM (*notice to airmission*), ou messages aux navigants aériens. Ces NOTAM contiennent des informations diverses, par exemple les aéroports, les pistes d'atterrissages disponibles, leur longueur, etc. Ces données sont utilisées pour prendre des décisions sûres et fiables en temps réel, un enjeu majeur pour garantir la sécurité et l'efficacité des vols.

L'intelligence artificielle symbolique, et notamment le raisonnement basé sur des ontologies [3], offre une approche robuste pour structurer ces données et réaliser des tâches complexes telles que :

- La classification (recatégorisation automatique d'entités, e.g. "piste trop courte" ou "mission satisfaisable", selon des critères prédéfinis),
- La détection d'inconsistances dans des bases de connaissances,
- La mise à jour incrémentale des bases suite à l'ajout ou la suppression de nouvelles entités (Abox), l'ontologie de base restant fixe (Tbox), e.g. en vol.

Plusieurs fragments disponibles dans la logique descriptive permettent d'adapter les problématiques étudiées grâce à leur simplicité en terme d'expressivité et d'efficacité algorithmique. C'est le cas du fragment OWL 2 EL, par exemple, pour lequel il existe une famille dominante d'algorithmes de raisonnement dits *consequence-based*, comme ceux utilisés par les raisonneurs ELK [2] ou encore ELepHant [4]. Ceux-ci permettent de résoudre des problèmes de classification avec une complexité polynomiale déterministe. Ces solutions sont aujourd'hui reconnues pour leur complétude et leur capacité à manipuler efficacement des bases de connaissance volumineuses et dynamiques. La possibilité de raisonner de manière incrémentale, sans recalculer toutes les inférences du système mais en intégrant uniquement une mise à jour, est particulièrement intéressante pour une efficacité optimale sur système embarqué [1].

Sujet

Le stage vise à développer un raisonneur spécialisé pour la logique EL adapté aux besoins spécifiques de l’avionique embarquée. Le raisonneur devra :

- Garantir la correction (pour *soundness*) et la complétude (absence de fausses inférences ou d’inférence manquantes),
- Manipuler dynamiquement les données de l’ABox (assertions individuelles), la TBox (ensemble des axiomes terminologiques) restant fixe,
- Implémenter une approche incrémentale pour des mises à jour rapides (ajout ou suppression d’individus),
- Optimiser les performances pour des architectures matérielles modernes (TPU, GPU, FPGA), en garantissant des temps de traitement inférieurs à 5 secondes pour des mises à jour d’ontologies complexes.

Une attention particulière sera portée sur l’efficacité algorithmique grâce à l’utilisation de structures de données binaires, qui sont particulièrement adaptées à une parallélisation sur architectures matérielles spécialisées.

Le stage se déroulera en plusieurs phases :

1. **Mois 1** : État de l’art
 - Étude des raisonneurs existants pour la logique EL, notamment ELK [2] et ELEPHANT [4], basés principalement sur le *consequence-based*.
 - Analyse des méthodes modernes d’implémentation parallèle sur architectures TPU/GPU/NPU.
 - Appropriation de l’approche binaire proposée pour optimiser la manipulation des données ontologiques.
2. **Mois 2** : Parsing d’ontologie et prototype initial
 - Implémentation d’un parseur d’ontologie basé sur des standards existants (e.g., OWL/XML) et sur un code existant (e.g. de ELEPHANT),
 - Développement d’une première version naïve du raisonneur utilisant l’approche binaire.
3. **Mois 3-4** : Implémentation parallèle
 - Révision de l’algorithme pour intégrer une parallélisation efficace adaptée aux architectures modernes.
 - Test et validation des performances sur des simulateurs matériels.
4. **Mois 5** : Optimisation incrémentale
 - Intégration de méthodes incrémentales pour gérer les modifications dynamiques de l’ABox (ajouts ou suppressions en vol) [1].
 - Validation de la complétude et de la *soundness* du raisonnement dans des scénarios avioniques réalistes.
5. **Mois 6** : Tests et finalisation
 - Tests sur une architecture matérielle réelle (e.g., tablette iPad Pro 2022 équipée d’un GPU moderne).
 - Documentation du code, rédaction du rapport et préparation de la présentation finale.

Références

- [1] Yevgeny Kazakov and Pavel Klinov. Incremental reasoning in owl el without bookkeeping. In *International semantic web conference*, pages 232–247. Springer, 2013.
- [2] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Elk reasoner : architecture and evaluation. In *ORE*. Citeseer, 2012.
- [3] Sebastian Rudolph. Foundations of description logics. In *Reasoning Web International Summer School*, pages 76–136. Springer, 2011.
- [4] Baris Sertkaya. The elephant reasoner system description. In *ORE*, pages 87–93, 2013.