

Common Lisp : une alternative à R pour les statistiques

Mots clés :

- **Directeur de thèse** : Carlos Agon
- **Co-encadrant(s)** :
- **Unité de recherche** : Sciences et Technologies de la Musique et du Son
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

1 Contexte 1.1 Cadre global R est un langage de programmation pour les statistiques, conçu initialement pour combiner les avantages de deux autres langages de programmation de l'époque : S (deja orienté statistiques) et Scheme (un dialecte de Lisp). Depuis sa conception, R a connu un grand succès pour finalement devenir le langage le plus utilisé par les statisticiens. Cependant, R fini aujourd'hui par montrer ses limites : il ne passe pas à l'échelle pour les applications modernes, gourmandes en quantité de données et en performances, et les limites de son expressivité le rendent inadapté à la diversité des applications récentes. 1.2 Enjeux L'idée de faire évoluer le langage lui-même n'est pas réaliste, car la communauté de ses utilisateurs est très conservatrice et réfractaire à la notion d'expérimentation. Il existe donc un enjeu majeur aujourd'hui, qui est de fournir une alternative à R, à même de satisfaire les demandes modernes de tous ses utilisateurs actuels et futurs. Cette alternative doit faire table rase du passé afin de repartir sur des bases plus modernes et saines. 2 Objectif L'objectif de ce travail consiste en la conception et l'implémentation d'une alternative à R, basée sur Common Lisp. Ce choix est justifié par un grand nombre d'arguments., à commencer par le simple fait qu'il soit aussi le choix de prédilection de l'auteur même de R. 2.1 Common Lisp Premièrement, R est originellement conçu à partir d'un noyau Scheme. Il est donc très proche de la famille des langages Lisp. Deuxièmement, de nombreuses lacunes ou faiblesses du noyau de R sont immédiatement corrigées par le simple fait de passer à Lisp. Troisièmement, dans une optique d'étude de haut niveau sur l'expressivité, Common Lisp, en tant que langage multi-paradigme par essence, est un très bon candidat. Enfin, dans une perspective de projet à portée industrielle, Common Lisp est un langage particulièrement bien adapté au développement de systèmes à grande échelle.

Résumé du projet de recherche (Langue 2)

2.2 Apports En dépit de tous les avantages prospectifs que l'utilisation de Common Lisp présente, force est de constater qu'à l'heure actuelle, aucun retour d'expérience concret n'a fait l'objet d'une véritable formalisation académique. — Common Lisp est un langage très utilisé dans l'industrie mais les industriels ne publient pas (ou très peu). — La famille des langages Lisp fait l'objet de recherches académiques, mais lorsque cette recherche est produite, elle est rarement basée sur Common Lisp et en général très orientée langage, c'est-à-dire de corrélation des applications industrielles. — Enfin, il existe de la recherche fondamentale dans des domaines où Common Lisp est a priori un excellent candidat pour l'expérimentation, mais force est de constater que même dans ces domaines, Common Lisp est systématiquement absent de la littérature. Le travail de thèse propose ici vise donc à corriger ce déséquilibre et, ce faisant, apporter une contribution majeure et novatrice à nos connaissances en matière de langages de programmation, en fournissant des éléments de réponse à la question générale suivante : qu'est-ce que le mélange de paradigmes apporte aux applications devant concilier performance et généralité, et où, dans cette optique se situent les avantages et les limites desdits paradigmes et de leurs interfaces ? 3 Organisations des travaux de recherche Il est prévu que le travail se déroule de la manière suivante. Étape 1 : Mise en place d'une base de travail Dans un premier temps, le travail consistera en la mise en place d'un cadre de base pour une étude comparative de R et Common Lisp : une application concrète d'un problème de statistiques pour la finance. Cette application sera développée d'au moins quatre manières différentes : 1. une version R idiomatique, naturelle et conforme à l'esprit du langage, 2. une version R dédiée, optimisée pour la performance au détriment de la généralité, 3. une version Common Lisp idiomatique tout en visant la reproduction des fonctionnalités de R, 4. une version Common Lisp dédiée. Étape 2 : Étude prospective sur les performances Une fois cette base de travail établie, nous pouvons déjà nous faire une idée des résultats auxquels on peut s'attendre en termes de performance, et mesurer l'impact de chaque décision de conception / implémentation sur celles-ci. Cette étape peut éventuellement être agrémente par de nouvelles répliques du problème de départ en C (pour une référence de performance très haute) et Julia (puisque c'est une solution concurrente récente à ne pas négliger). Étape 3 : Étude paradigmatique À partir de l'analyse précedente, de nombreuses alternatives s'offrent à nous, consistant à changer de paradigme(s) pour exprimer les mêmes idées. Dans cette étape, il conviendra d'étudier les très grandes possibilités offertes par Common Lisp, ainsi que leur influence sur les performances et sur l'expressivité offerte aux programmeurs. Étape 4 : à définir. . . La somme des questions qui vont apparaître à ce niveau est trop élevée pour ne faire l'objet que d'une thèse en trois ans. En fonction des discussions qui vont émerger, des priorités de chaque acteur du projet ainsi que des préférences du candidat, un ensemble de problèmes plus restreint sera choisi et développé en profondeur dans la suite du travail.

