

### Conclusion Générale

Les méthodes formelles sont de plus en plus appelées pour la vérification de programmes écrits dans des langages de programmations modernes. Cette philosophie originaire des idées du projet JPF (Java PathFinder) lancé par la NASA en 1999 pour analyser des programmes Java. L'analyser ce fait par l'application du Spin model-checker indirectement sur les programmes Java, en passant par une étape intermédiaire de translation vers le Promela, cette phase est nécessaire étant donné que le Spin accepte les inputs dans cette langue.

Ces mêmes idées ont été reprises par extension à d'autre langage de programmation. Vu que le langage C est l'un des langages de programmations les plus populaires sollicité dans le développement des programmes dans divers domaines. Plusieurs efforts ont été investi afin de mettre en œuvre des outils de translation complets du C vers Promela, néant moins tous les attentives souffrent de limitations. Ces dernières sont dues en leur globalité à la richesse et la diversité du langage C comparant avec le Promela.

La translation proposée dans notre projet, suit le principe de la translation dirigée par la grammaire, où nous avons augmenté cette dernière avec des attributs pour mieux contrôler le processus et l'arbre d'analyse abstrait. Cependant, nous avons introduit une technique de sélection des chemins grammaticaux qui nous évite d'élaborer des chemins sans intérêt, et de ce fait ne génère que ce qui est sûr. Une fois on assure que le chemin est unique, la translation est engagée immédiatement vers le Promela. Cette façon de faire optimise considérablement le temps d'exécution du translateur baptisé CProl. Il assure la translation correcte de tout le sous-ensemble C choisi.

Dans notre travail nous avons essayé de surmonter certaines des limitations déjà évoquées, à savoir : une proposition d'implémenter une structure de données composées non supportée par le Promela qui sont les tableaux multidimensionnels pour rendre la translation des tableaux multidimensionnels naturellement supportés en C, facile à réaliser.

Au niveau des appels de fonctions récursives qui ne sont pas prises en charge d'une façon complète, surtout que le mécanisme du C et Promela sont différent, puisque en Promela les fonctions sont modélisées par des processus où la valeur retournée est envoyée et reçue par des canaux. Ce mécanisme limite notre souplesse de passage de paramètre si facile en C. pour surmonter cette difficulté nous avons tenté de manipuler soigneusement le mécanisme des canaux.

Nous avons profité de l'atomicité existante en Promela pour assurer que certaines séquences du code C s'exécutent d'une façon indivisible.

Le développement du Prototype CProl a été réalisé suivant une vision orienté objet avec le C #.

### Perspectives et travaux futurs

Automatisation du lancement du test qui nécessite la translation du code, gérer l'initialisation suivi du lancement du Spin.

Les limitations principales sont les pointeurs du C, plusieurs actuellement sont engagés pour y trouver une solution concluante. Cependant aucune d'entre elle n'est complète.

Nous voudrions attiré l'attention sur les bibliothèque parallèles du C, spécialement la bibliothèque Wool et OpenMP( Open Multiprocessing).qui contiennent des fonctions assurant la concurrence, une notion sensible et délicate en programmation. Les algorithmes concurrentiels sont exposés à des erreurs pouvant ne pas être détectées au niveau design. Donc, il serait très intéressant de les tester avec le Spin surtout qu'il est dédié pour la vérification u parallélisme dans les modèles.