

The Size-Change Principle for Program Termination (notes)

1^{er} février 2012

Résumé

Les auteurs de cet article apportent une nouvelle méthode basée sur le changement de taille des paramètres lors d'appels de fonction pour détecter la terminaison de programme. Ils assimilent le graphe d'appel du programme à un automate de Büchi grâce auquel ils vont pouvoir détecter les décroissances des paramètres.

1 Idée générale

Ils considèrent des séquences d'appels de fonction infinies suivant le graphe de flot de contrôle du programme. Le raisonnement est similaire à celui de foetus mais offre une plus grande aire de détection. Par exemple, la fonction `zip`¹ nécessite une réécriture pour être détectée terminante par foetus car l'ordre des paramètres est important (diagonale de la matrice d'appel). Dans cette nouvelle méthode, l'ordre des paramètres importe peu car seule la décroissance des valeurs en paramètres est déterminante.

Sous ces hypothèses,

- ordre bien fondé sur les données,
- le *multipath*² est donné.

ils appliquent le raisonnement suivant : si tous les chemins d'exécution de longueur infinie produisent une séquence de valeurs décroissantes alors en utilisant les hypothèses, aucune exécution infinie n'est possible.

2 Inconvénient de cette méthode

Pour déterminer si tous les chemins d'exécution infinis produisent une séquence de valeurs décroissantes, ils créent deux classes de chemins. La première $FLOW^\omega$, elle contient tous les flots de contrôle possibles de longueur infinie. La seconde $DESC^\omega$, celle-ci contient seulement les flots de contrôle dont les séquences sont décroissantes. Pour montrer qu'un programme termine, il faut déterminer l'égalité des ensembles $FLOW^\omega$ et $DESC^\omega$.

1. fonction présente dans le papier foetus.

2. La construction précise de cette structure est un problème indécidable, on devrait donc en cas d'implémentation se limiter à une approximation.

Ce problème se ramenant à montrer l'équivalence de deux automates de Büchi est connu et est classé dans PSPACE. Néanmoins ils proposent un autre algorithme basé sur les graphes mais dont ils ramènent l'étude à un problème PSPACE-hard.

Ils concluent donc que l'utilisation du critère de changement de taille de paramètre est PSPACE-hard.

3 Conclusion

La mise en oeuvre de cet algorithme n'est donc pas envisageable dans le cas général. Les auteurs affirment qu'ils travaillent sur certaines restrictions permettant de réduire la complexité en temps (cubique). Malheureusement l'une de ces restrictions consistent à ne pas pouvoir modifier la place des paramètres.