Vérification des programmes (probabilistes) d'ordre supérieur

Charles Grellois
Equipe LIRICA

Aix-Marseille Université

Visite du COS
12 mars 2020

Cet exposé

Je présente succintement mes travaux sur la vérification des programmes d'ordre supérieur depuis mon arrivée à LIRICA (septembre 2017) :

- Pierre Clairambault, Charles Grellois, Andrzej S. Murawski: Linearity in higher-order recursion schemes. POPL 2018
- Naoki Kobayashi, Ugo Dal Lago, Charles Grellois: On the Termination Problem for Probabilistic Higher-Order Recursive Programs. LICS 2019
- Ugo Dal Lago, Charles Grellois: Probabilistic Termination by Monadic Affine Sized Typing. ACM TOPLAS 2019 (version longue de ESOP 2017)

Programmes d'ordre supérieur, programmes probabilistes

- D'ordre supérieur : une fonction peut prendre en argument des fonctions, qui prennent en argument des fonctions...

$$\texttt{map}\ \varphi\ [0,1,2] \qquad \text{retourne} \qquad [\varphi(0), \varphi(1), \varphi(2)].$$

- Probabiliste : selon une certaine probabilité, un programme va effectuer une action ou une autre

$$\begin{array}{lcll} M \oplus_p N & \to & M & \text{avec prob. } p \\ & \to & N & \text{avec prob. } 1-p \end{array}$$

Vérifier les programmes d'ordre supérieur

Plusieurs approches, dont :

- Le model-checking : on approxime le programme en un modèle, et on fait de la vérification sur le modèle selon une méthode systématique
- La théorie des types : on n'approxime pas le programme, mais on l'annote, si on y arrive, par des informations permettant la vérification

On va regarder un peu ces deux approches pour la terminaison probabiliste.

Mais avant, le cas déterministe pour un problème plus complexe de vérification.

Model-checking déterministe des programmes d'ordre supérieur

Model-checking d'ordre supérieur (HOMC) : modéliser un programme fonctionnel comme un schéma de récursion d'ordre supérieur.

$$\begin{array}{rcl} \texttt{Main} & = & \texttt{Listen Nil} \\ \texttt{Listen}\ x & = & \texttt{if end\_signal() then}\ x \\ & & \texttt{else Listen received\_data()} :: x \end{array}$$

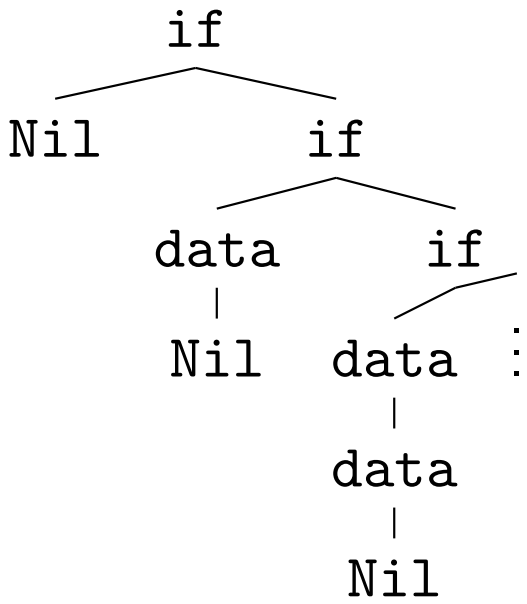
est approximé par le schéma de récursion

$$\mathcal{G} \quad = \quad \begin{cases} \texttt{S} & = & \texttt{L Nil} \\ \texttt{L}\ x & = & \texttt{if}\ x\ (\texttt{L}\ (\texttt{data}\ x\ )\ ) \end{cases}$$

Model-checking déterministe des programmes d'ordre supérieur

$$\mathcal{G} \quad = \quad \begin{cases} \mathtt{S} & = & \mathtt{L\ Nil} \\ \mathtt{L}\ x & = & \mathtt{if}\ x\ (\mathtt{L}\ (\mathtt{data}\ x\ )\ ) \end{cases}$$

qui représente l'arbre infini d'actions suivant (obtenu par réécriture infinitaire)



Model-checking déterministe des programmes d'ordre supérieur

Logique monadique du second ordre (MSO) :

- équivalente au mu-calcul modal sur les arbres
- contient LTL, CTL...

Le model-checking de MSO sur les arbres infinis générés par les schémas :

- est décidable (Ong 2006, Kobayashi et Ong 2009,...)
- complexité : n-EXPTIME pour n l'ordre du schéma (Ong 2006,...)
- mieux : n-EXPTIME pour n l'ordre linéaire du schéma (une des contributions de POPL 2018)

Analyse de terminaison à la HOMC

On étend les schémas (HORS) en des schémas probabilistes (PHORS).
Exemple : marche aléatoire :

$$\mathcal{G} \quad = \quad \begin{cases} \mathtt{S} & = & \mathtt{F\ e} \\ \mathtt{F}\ x & = & x \oplus_p \mathtt{F\ (\ F}\ x\ \mathtt{)} \end{cases}$$

LICS 2019 : terminaison quasi-sûre (AST, avec proba $= 1$) : indécidable pour l'ordre 2 et au-delà.

Réduction du problème à l'ordre n à des calculs de points fixes à l'ordre n-1.

Procédure fiable d'approximation des probabilités de terminaison (plus général que AST).

Problème : indécidable en général, mais approximable de façon incomplète à l'ordre 2 (et plus ?).

Types et terminaison probabiliste

ESOP 2017, version longue ACM TOPLAS 2019 :

Pour une fonction récursive telle que (encore la marche aléatoire !) :

$$\begin{array}{rcccl} f & : & n+1 & \mapsto & n \oplus_{\frac{1}{2}} f(n+2) \\ & & 0 & \mapsto & 0 \end{array}$$

on donne un système de types (= d'annotations des programmes) donnant des informations sur la taille des arguments, et vérifiant que les appels récursifs induisent un système markovien convergeant vers 0 avec probabilité 1.

Typable $\Rightarrow$ presque sûrement terminant (= terminant avec probabilité 1).

Mais peu de programmes typables. . . il faut en particulier être affine.

Conclusions et perspectives

Conclusions:

- Mesure de la complexité plus fine pour le model-checking déterministe
- Premières approches pour la terminaison des schémas probabilistes
- Un système de types (très incomplet) pour la terminaison quasi-sûre des programmes probabilistes

Quelques perspectives:

- Model-checking de LTL pour les schémas probabilistes
- Un mu-calcul modal probabiliste pour vérifier les schémas probabilistes ?
- Etendre l'algorithme d'approximation pour la terminaison des schémas probabilistes
- Un système de types plus complet pour la terminaison quasi-sûre des programmes probabilistes