Structuring the Synthesis of Heap-Manipulating Programs

NADIA POLIKARPOVA, UCSD, USA
ILYA SERGEY, University College London, UK

Introduction

$\{x \mapsto a * y \mapsto b\}$ **void** swap(**loc** x, **loc** y) $\{x \mapsto b * y \mapsto a\}$

Motivation : Faire avancer l'état de l'art en matiére in synthesizing provably correct heap-manipulating programs from declarative functional specifications

Spécifications pour la Synthèse

Règles d'Inférence Basiques

Unification Spatiale et Backtrack

Raisonner sur les contraintes pures

Préconditions

Raisonner sur les contraintes pures

Postconditions

Synthèse pour prédicats inductifs

Mémoire dynamique

Synthèse pour prédicats inductifs

Induction

Synthèse pour prédicats inductifs

Déroulement de prédicat

Synthèse pour prédicats inductifs

Etiquette de niveau

Synthèse pour prédicats inductifs

Déroulement dans la postcondition

Permettre l'appel de procèdure

Enlévement de l'appel

Synthetic Separation Logic

Garanties Formelles

Algorithme de synthèse basé sur SSL

Optimisations et extensions

Optimisations :

- Règles inversibles

Optimisations et extensions

Optimisations :

- Règles inversibles
- Recherche multi-phase

Optimisations et extensions

Optimisations :

- Règles inversibles
- Recherche multi-phase
- Rèduction des symétries

Optimisations et extensions

Optimisations :

- Règles inversibles
- Recherche multi-phase
- Rèduction des symétries
- Règles d'échec

Optimisations et extensions

Optimisations :

- Règles inversibles
- Recherche multi-phase
- Rèduction des symétries
- Règles d'échec

Extensions :

- Fonctions auxilliaire

Optimisations et extensions

Optimisations :

- Règles inversibles
- Recherche multi-phase
- Rèduction des symétries
- Règles d'échec

Extensions :

- Fonctions auxilliaire
- Enlèvement de branches

Benchmark