

# Tackling performance issues of symbolic execution in CPAchecker (naming of CPA's has to be discussed)

April 28, 2015

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>   | <b>3</b> |
| 1.1      | Motivation . . . . .  | 3        |
| 1.2      | Existing problems of symbolic value analysis (short overview)                                 | 3        |
| <b>2</b> | <b>Theoretical background</b>   | <b>3</b> |
| 2.1      | General Overview of configurable program analysis . . . . .                                   | 3        |
| 2.1.1    | Configurable program analysis definition . . . . .  | 3        |
| 2.1.2    | Location CPA as example CPA . . . . .   | 3        |
| 2.1.3    | CPAchecker as a framework for configurable program<br>analysis . . . . .                      | 3        |
| 2.2      | Basic definition of CPAs used in this paper . . . . .   | 3        |
| 2.2.1    | Symbolic Value Analysis CPA . . . . .   | 3        |
| 2.2.2    | Constraints CPA . . . . .   | 3        |
| 2.2.3    | Composition of Location CPA, Symbolic Value Anal-<br>ysis CPA and Constraints CPA . . . . .   | 3        |
| <b>3</b> | <b>Implementation of these CPAs in CPAchecker</b>   | <b>3</b> |
| 3.1      | Basic implementation . . . . .  | 3        |
| 3.1.1    | LocationCPA already existed, implemented as defined   | 3        |
| 3.1.2    | Symbolic Value Analysis as extension to existing Val-<br>ueAnalysisCPA . . . . .              | 3        |
| 3.1.3    | ConstraintsCPA as new CPA . . . . .   | 3        |
| 3.2      | Existing options/optimizations . . . . .  | 3        |
| 3.2.1    | Constraints State simplifications . . . . .   | 3        |
| 3.2.1.1  | Computing definite assignments for symbolic<br>values (+ strengthening in value analysis) . . | 3        |

|          |  |          |
|----------|--|----------|
| 3.2.1.2  | Handling of trivial constraints - completely trivial ones not even added to state, ones with definite assignment removed after sat check . . . . . | 3        |
| 3.2.1.3  | . . . . .  | 3        |
| 3.2.2    | Different merge operators . . . . .  | 4        |
| 3.2.2.1  | merge sep . . . . .  | 4        |
| 3.2.2.2  | merge join . . . . .   | 4        |
| 3.2.3    | Different less-or-equal operators . . . . .  | 4        |
| <b>4</b> | <b>Performance issues of implementation</b>  | <b>4</b> |
| 4.1      | SAT checks . . . . .   | 4        |
| 4.2      | Path explosion . . . . .   | 4        |
| <b>5</b> | <b>CEGAR as general means of speed up</b>  | <b>4</b> |

# 1 Introduction

## 1.1 Motivation

## 1.2 Existing problems of symbolic value analysis (short overview)

# 2 Theoretical background

## 2.1 General Overview of configurable program analysis

### 2.1.1 Configurable program analysis definition

### 2.1.2 Location CPA as example CPA

### 2.1.3 CPAchecker as a framework for configurable program analysis

## 2.2 Basic definition of CPAs used in this paper

### 2.2.1 Symbolic Value Analysis CPA

### 2.2.2 Constraints CPA

### 2.2.3 Composition of Location CPA, Symbolic Value Analysis CPA and Constraints CPA

# 3 Implementation of these CPAs in CPAchecker

## 3.1 Basic implementation

### 3.1.1 LocationCPA already existed, implemented as defined

### 3.1.2 Symbolic Value Analysis as extension to existing Value-AnalysisCPA

### 3.1.3 ConstraintsCPA as new CPA

## 3.2 Existing options/optimizations

### 3.2.1 Constraints State simplifications

#### 3.2.1.1 Computing definite assignments for symbolic values (+ strengthening in value analysis)

#### 3.2.1.2 Handling of trivial constraints - completely trivial ones not even added to state, ones with definite assignment removed after sat check

#### 3.2.1.3

### **3.2.2 Different merge operators**

#### **3.2.2.1 merge sep**

#### **3.2.2.2 merge join**

### **3.2.3 Different less-or-equal operators**

## **4 Performance issues of implementation**

### **4.1 SAT checks**

### **4.2 Path explosion**

## **5 CEGAR as general means of speed up**