

---

## Contents

<b>Preface</b> .....	vii
<b>Contributions</b> .....	xvii
<b>List of Figures</b> .....	xix
<b>1 Introduction</b> .....	1
1.1 Technical Background .....	2
1.2 Value-Range Analysis .....	4
1.3 Analysing C .....	6
1.4 Soundness .....	7
1.4.1 An Abstraction of C .....	7
1.4.2 Combining Value and Content Abstraction .....	8
1.4.3 Combining Pointer and Value-Range Analysis .....	9
1.5 Efficiency .....	11
1.6 Completeness .....	15
1.6.1 Analysing String Buffers .....	16
1.6.2 Widening with Landmarks .....	16
1.6.3 Refining Points-to Analysis .....	17
1.6.4 Further Refinements .....	17
1.7 Related Tools .....	18
1.7.1 The Astrée Analyser .....	18
1.7.2 SLAM and ESPX .....	19
1.7.3 CCured .....	20
1.7.4 Other Approaches .....	20
<b>2 A Semantics for C</b> .....	23
2.1 Core C .....	23
2.2 Preliminaries .....	28
2.3 The Environment .....	28
2.4 Concrete Semantics .....	32

2.5	Collecting Semantics .....	37
2.6	Related Work .....	42

---

**Part I Abstracting Soundly**

---

<b>3</b>	<b>Abstract State Space .....</b>	<b>47</b>
3.1	An Introductory Example .....	48
3.2	Points-to Analysis .....	51
3.2.1	The Points-to Abstract Domain .....	54
3.2.2	Related Work .....	55
3.3	Numeric Domains .....	56
3.3.1	The Domain of Convex Polyhedra .....	56
3.3.2	Operations on Polyhedra .....	59
3.3.3	Multiplicity Domain .....	62
3.3.4	Combining the Polyhedral and Multiplicity Domains ..	65
3.3.5	Related Work .....	68
<b>4</b>	<b>Taming Casting and Wrapping .....</b>	<b>71</b>
4.1	Modelling the Wrapping of Integers .....	72
4.2	A Language Featuring Finite Integer Arithmetic .....	74
4.2.1	The Syntax of Sub C .....	74
4.2.2	The Semantics of Sub C .....	75
4.3	Polyhedral Analysis of Finite Integers .....	76
4.4	Implicit Wrapping of Polyhedral Variables .....	77
4.5	Explicit Wrapping of Polyhedral Variables .....	78
4.5.1	Wrapping Variables with a Finite Range .....	78
4.5.2	Wrapping Variables with Infinite Ranges .....	80
4.5.3	Wrapping Several Variables .....	80
4.5.4	An Algorithm for Explicit Wrapping .....	82
4.6	An Abstract Semantics for Sub C .....	83
4.7	Discussion .....	86
4.7.1	Related Work .....	87
<b>5</b>	<b>Overlapping Memory Accesses and Pointers .....</b>	<b>89</b>
5.1	Memory as a Set of Fields .....	89
5.1.1	Memory Layout for Core C .....	90
5.2	Access Trees .....	93
5.2.1	Related Work .....	99
5.3	Mixing Values and Pointers .....	100
5.4	Abstraction Relation .....	106
5.4.1	On Choosing an Abstraction Framework .....	108

<b>6</b>	<b>Abstract Semantics</b> .....	111
6.1	Expressions and Simple Assignments .....	116
6.2	Assigning Structures .....	118
6.3	Casting, &-Operations, and Dynamic Memory .....	121
6.4	Inferring Fields Automatically .....	123

---

## Part II Ensuring Efficiency

---

<b>7</b>	<b>Planar Polyhedra</b> .....	127
7.1	Operations on Inequalities .....	129
7.1.1	Entailment between Single Inequalities .....	130
7.2	Operations on Sets of Inequalities .....	131
7.2.1	Entailment Check .....	131
7.2.2	Removing Redundancies .....	132
7.2.3	Convex Hull .....	134
7.2.4	Linear Programming and Planar Polyhedra .....	144
7.2.5	Widening Planar Polyhedra .....	145
<b>8</b>	<b>The TVPI Abstract Domain</b> .....	147
8.1	Principles of the TVPI Domain .....	148
8.1.1	Entailment Check .....	150
8.1.2	Convex Hull .....	150
8.1.3	Projection .....	151
8.2	Reduced Product between Bounds and Inequalities .....	152
8.2.1	Redundancy Removal in the Reduced Product .....	155
8.2.2	Incremental Closure .....	156
8.2.3	Approximating General Inequalities .....	160
8.2.4	Linear Programming in the TVPI Domain .....	160
8.2.5	Widening of TVPI Polyhedra .....	161
8.3	Related Work .....	163
<b>9</b>	<b>The Integral TVPI Domain</b> .....	165
9.1	The Merit of $\mathbb{Z}$ -Polyhedra .....	166
9.1.1	Improving Precision .....	166
9.1.2	Limiting the Growth of Coefficients .....	167
9.2	Harvey's Integral Hull Algorithm .....	168
9.2.1	Calculating Cuts between Two Inequalities .....	169
9.2.2	Integer Hull in the Reduced Product Domain .....	172
9.3	Planar $\mathbb{Z}$ -Polyhedra and Closure .....	177
9.3.1	Possible Implementations of a $\mathbb{Z}$ -TVPI Domain .....	177
9.3.2	Tightening Bounds across Projections .....	179
9.3.3	Discussion and Implementation .....	180
9.4	Related Work .....	182

<b>10</b>	<b>Interfacing Analysis and Numeric Domain</b>	185
10.1	Separating Interval from Relational Information	185
10.2	Inferring Relevant Fields and Addresses	187
10.2.1	Typed Abstract Variables	189
10.2.2	Populating the Field Map	190
10.3	Applying Widening in Fixpoint Calculations	192

---

## Part III Improving Precision

---

<b>11</b>	<b>Tracking String Lengths</b>	197
11.1	Manipulating Implicitly Terminated Strings	198
11.1.1	Analysing the String Loop	199
11.1.2	Calculating a Fixpoint of the Loop	203
11.1.3	Prerequisites for String Buffer Analysis	209
11.2	Incorporating String Buffer Analysis	209
11.2.1	Extending the Abstraction Relation	212
11.3	Related Work	213
<b>12</b>	<b>Widening with Landmarks</b>	217
12.1	An Introduction to Widening/Narrowing	217
12.1.1	The Limitations of Narrowing	218
12.1.2	Improving Widening and Removing Narrowing	220
12.2	Revisiting the Analysis of String Buffers	220
12.2.1	Applying the Widening/Narrowing Approach	222
12.2.2	The Rationale behind Landmarks	222
12.2.3	Creating Landmarks for Widening	225
12.2.4	Using Landmarks in Widening	225
12.3	Acquiring Landmarks	226
12.4	Using Landmarks at a Widening Point	227
12.5	Extrapolation Operator for Polyhedra	229
12.6	Related Work	231
<b>13</b>	<b>Combining Points-to and Numeric Analyses</b>	235
13.1	Boolean Flags in the Numeric Domain	237
13.1.1	Boolean Flags and Unbounded Polyhedra	238
13.1.2	Integrality of the Solution Space	239
13.1.3	Applications of Boolean Flags	240
13.2	Incorporating Boolean Flags into Points-to Sets	241
13.2.1	Revising Access Trees and Access Functions	241
13.2.2	The Semantics of Expressions and Assignments	244
13.2.3	Conditionals and Points-to Flags	246
13.2.4	Incorporating Boolean Flags into the Abstraction Relation	249

13.3	Practical Implementation . . . . .	250
13.3.1	Inferring Points-to Flags on Demand . . . . .	251
13.3.2	Populating the Address Map on Demand . . . . .	251
13.3.3	Index-Sensitive Memory Access Functions . . . . .	253
13.3.4	Related Work . . . . .	255
<b>14</b>	<b>Implementation . . . . .</b>	<b>259</b>
14.1	Technical Overview of the Analyser . . . . .	260
14.2	Managing Abstract Domains . . . . .	262
14.3	Calculating Fixpoints . . . . .	264
14.3.1	Scheduling of Code without Loops . . . . .	265
14.3.2	Scheduling in the Presence of Loops and Function Calls . . . . .	267
14.3.3	Deriving an Iteration Strategy from Topology . . . . .	268
14.3.4	Related Work . . . . .	269
14.4	Limitations of the String Buffer Analysis . . . . .	271
14.4.1	Weaknesses of Tracking First NUL Positions . . . . .	271
14.4.2	Handling Symbolic NUL Positions . . . . .	272
14.5	Proposed Future Refinements . . . . .	276
<b>15</b>	<b>Conclusion and Outlook . . . . .</b>	<b>277</b>
	<b>A Core C Example . . . . .</b>	<b>281</b>
	<b>References . . . . .</b>	<b>285</b>
	<b>Index . . . . .</b>	<b>297</b>

<http://www.springer.com/978-1-84800-016-2>

Value-Range Analysis of C Programs  
Towards Proving the Absence of Buffer Overflow  
Vulnerabilities

Simon, A.

2008, XXII, 302 p. 119 illus., Hardcover

ISBN: 978-1-84800-016-2