

Subseries of Lecture Notes in Computer Science

Edited by J. Siekmann

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis



J. G. Williams

Instantiation Theory

On the Foundations of Automated Deduction

Springer-Verlag

Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editor

Jörg Siekmann

Institut für Informatik, Universität Kaiserslautern

Postfach 3049, W-6750 Kaiserslautern, FRG

Author

James G. Williams

The MITRE Corporation, M/S A129

Burlington Road, Bedford, MA 01730-0208, USA

CR Subject Classification (1991): I.2.3

ISBN 3-540-54333-3 Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-54333-3 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this publication or parts thereof is only permitted under the provisions of the German Copyright Law of September 9, 1965, in its current version, and a copyright fee must always be paid. Violations fall under the prosecution act of the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1991

Printed in Germany

Typesetting: Camera ready by author

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.

2145/3140-543210 - Printed on acid-free paper

PREFACE

Instantiation theory is the study of instantiation in an abstract context that is applicable to most commonly studied logical formalisms. This book begins with a survey of general approaches to the study of instantiation, as found in tree systems, order-sorted algebras, algebraic theories, composita, and *instantiation systems*.

A classification of instantiation systems is given, based on properties of substitutions, degree of type strictness, and well-foundedness of terms. Equational theories and the use of typed variables are studied in terms of quotient homomorphisms and embeddings, respectively. Every instantiation system is a quotient system of a subsystem of first-order term instantiation.

A general unification algorithm is developed as an application of the basic theory. Its soundness is rigorously proved, and its completeness and efficiency are verified for certain classes of instantiation systems. Appropriate applications of the algorithm include unification of first-order terms, order-sorted terms, and first-order formulas modulo α -conversion, as well as equational unification using simple congruences.

I am indebted to William Farmer for acquainting me with the literature on unification algorithms, for help in formulating the basic theory, and for valuable advice regarding its development. I also wish to thank Hans-Jürgen Bärkert, Dale Johnson, John Stell, and an anonymous referee for valuable feedback on its presentation. This work was sponsored by the Rome Laboratories, Griffiss Air Force Base, Rome, NY 13441, under the direction of John C. Faust, COAC.

Bedford, MA
June 1991

James G. Williams

TABLE OF CONTENTS

Section	Title	Page
1	Introduction	1
	Motivation	1
	Organization and Content	1
	Related Work	2
2	Background	4
	Monoid Actions	4
	The Unification Problem	7
	Motivating Examples	9
3	General Approaches to Instantiation	11
	Instantiation Systems	11
	Tree Instantiation Systems	17
	Order-sorted Instantiation	18
	Other Systems of Historical Interest	19
4	Classification Properties	23
	Variable Dependency Properties	23
	Type Strictness Properties	25
	Term Occurrence Properties	26
	Iteration and Fixed Points	28
	Counterexamples	33
5	Homomorphisms	37
	Congruences and Quotient Systems	37
	Subsystems and Embeddings	40
	Arbitrary Homomorphisms	45
	Counterexamples	47
6	Construct Bases	49
	Basic Properties of Constructs	51
	Finite Tree Implementations	52
	Term Weights	55
	Examples	56

7	Unification — an Algorithm and its Soundness	61
	Requirements	61
	Basic Strategies	64
	Algorithmic Specification	65
	Soundness of the Algorithm	68
	Choice of Instantiation System	70
	Examples	75
8	Term-Implementation and Completeness	81
	Concrete Term Implementation	81
	Completeness	83
	Step-wise Derivation of unification_2	89
	Tree-Unification	92
	Counterexamples	94
9	Implementation and Computational Complexity	97
	Deferred Merging	98
	Fragment Merging	100
	Implementation of find_g	103
	Implementation of base_g	108
	Implementation of $\text{vdc}(E, g)$	112
	Further Optimizations	116
	Computational Complexity	117
10	Related Issues not Addressed	121
Appendix A	The Compiled unification_7 Specification	122
References		128