

# Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

139

---

J. Uhl S. Drossopoulou  
G. Persch G. Goos M. Dausmann  
G. Winterstein W. Kirchgässner

An Attribute Grammar  
for the Semantic  
Analysis of Ada

---



Springer-Verlag  
Berlin Heidelberg New York 1982

### **Editorial Board**

D.R. Barstow, W. Brauer P. Brinch Hansen D. Gries D.C. Luckham  
C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

### **Authors**

Jürgen Uhl  
Sophia Drossopoulou  
Guido Persch  
Gerhard Goos  
Manfred Dausmann  
Georg Winterstein  
Walter Kirchgässner  
Institut für Informatik II der Universität  
Postfach 6380, 7500 Karlsruhe 1, Germany

CR Subject Classifications (1979): 3.1, 3.2, 3.4

ISBN 3-540-11571-4 Springer-Verlag Berlin Heidelberg New York  
ISBN 0-387-11571-4 Springer-Verlag New York Heidelberg Berlin

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© by Springer-Verlag Berlin Heidelberg 1982  
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.  
2145/3140-543210

## P r e f a c e

=====

Attribute grammars are an established tool for the formal specification of the semantics of a programming language and also the specification of the language's compiler. This book contains an attribute grammar specifying the static semantics of Ada, together with an explanatory introduction.

This attribute grammar, which we shall call the AG, completely describes the semantics of Ada, as published in July 1980. The AG was extensively tested with the help of an equivalent automatically generated Pascal program. From this specification we systematically developed the semantic analysis part of our Ada compiler front-end.

Part A of this volume describes the development of the AG. It contains a survey of the tasks of semantic analysis within a front-end, the use of attribute grammars for the specification of static semantics and semantic analysis, and the procedure for writing the AG.

Part B is a rationale for the AG: the three main tasks of semantic analysis are described, the main attributes are introduced, their use, their dependencies and their types are outlined.

Part C contains the AG itself.

We are particularly indebted to Brigitte Hutt, Uwe Kastens and Erich Zimmermann, without whose support the present work could not have been carried out: They provided the tools for the systematic and efficient use of attribute grammars, and they readily and frequently advised us on the design of the AG.

This research was partially supported by the Bundesamt für Wehrtechnik und Beschaffung under Contract No. E/F61D/90104/95031.

Table of Contents  
=====

PART A : DEVELOPMENT OF THE AG  
=====

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| 1. Introduction and Survey                                                         | 3  |
| 1.1. Survey of the Karlsruhe Ada Compiler Front-End                                | 3  |
| 1.2. Tasks and Modularization of the Front-End                                     | 4  |
| 1.3. Tasks of Semantic Analysis                                                    | 5  |
| 1.4. Formal Specification of Semantic Analysis                                     | 7  |
| 1.5. Specification of Semantic Analysis with Attribute Grammars                    | 9  |
| 1.5.1. Attribute Grammars                                                          | 9  |
| 1.5.2. General Strategies in Specifying Static Semantics with an Attribute Grammar | 10 |
| 1.5.3. From Static Semantics Specification to Semantic Analysis Specification      | 10 |
| 1.5.4. Partitioned Attribute Grammars                                              | 11 |
| 1.5.5. GAG : Generator for Attribute Grammars                                      | 12 |
| 1.6. ALADIN : A Language for Attribute Grammar Definitions                         | 13 |
| 1.7. An Example                                                                    | 18 |
| 2. Procedure for the AG                                                            | 22 |
| 2.1. Semantic Analysis for Ada                                                     | 22 |
| 2.2. Diana and the AG                                                              | 23 |
| 2.3. Development of the AG                                                         | 24 |
| 2.4. Completeness and Parametrization of the AG                                    | 25 |
| 2.5. Implementation of the AG in Ada                                               | 26 |
| 2.6. Adaptibility to Ada'82                                                        | 26 |
| 3. Comparison of the AG with other Attribute Grammars                              | 27 |
| 4. Experience with Attribute Grammars                                              | 29 |

|                                                |    |
|------------------------------------------------|----|
| PART B: RATIONALE FOR THE AG<br>=====          | 31 |
| 5. Overall Structure of the AG                 | 35 |
| 5.1 Main Semantic Tasks                        | 35 |
| 5.2 Classes of Attributes                      | 37 |
| 6. Declaration Elaboration                     | 40 |
| 6.1. at_env_in, at_env_out                     | 41 |
| 6.2. at_keys                                   | 43 |
| 6.3. at_keys1, at_keys2, at_env                | 46 |
| 6.4. at_labels, at_loops_blocks                | 48 |
| 6.5. at_pos, at_char                           | 49 |
| 6.6. at_impl_descrs                            | 49 |
| 6.7. at_impl_subprs                            | 51 |
| 6.8. at_user_reps                              | 52 |
| 6.9. at_user_rep                               | 54 |
| 7. Name and Expression Analysis                | 56 |
| 7.1. at_allowed1                               | 57 |
| 7.2. at_with_use                               | 57 |
| 7.3. at_allowed2                               | 58 |
| 7.4. at_valid                                  | 59 |
| 7.5. at_expected                               | 59 |
| 7.6. at_context                                | 61 |
| 7.7. at_expected_in, at_expected_out           | 63 |
| 8. Semantic Checks and Error Handling          | 66 |
| 8.1. General Strategies                        | 66 |
| 8.1.1. Error Recognition                       | 66 |
| 8.1.2. Error Messages                          | 66 |
| 8.1.3. Error Recovery                          | 67 |
| 8.2. Classes of Errors                         | 68 |
| 8.3. Structural Errors                         | 70 |
| 8.3.1. at_positional_allowed                   | 70 |
| 8.3.2. at_is_last                              | 70 |
| 8.3.3. at_empty                                | 71 |
| 8.3.4. at_unconstrained                        | 71 |
| 8.3.5. at_has_range                            | 71 |
| 8.3.6. at_init                                 | 71 |
| 8.3.7. at_encl_unit                            | 71 |
| 8.3.8. at_return_required                      | 72 |
| 8.3.9. at_is_operator, at_param_no             | 72 |
| 8.3.10. at_block_no                            | 73 |
| 8.3.11. at_first_select, at_select_state       | 73 |
| 8.3.12. at_no_code_stms, at_no_decls, at_empty | 73 |
| 8.3.13. at_is_item                             | 73 |
| 8.3.14. at_is_exp                              | 73 |

|             |                                                         |    |
|-------------|---------------------------------------------------------|----|
| 8.4.        | Errors Concerning Declaration Elaboration               | 74 |
| 8.4.1.      | Completeness of Declarations                            | 74 |
| 8.4.2.      | Matching of Declarations                                | 74 |
| 8.4.3.      | Hiding                                                  | 75 |
| 8.4.4.      | Linearity of Declarations                               | 76 |
| 8.4.5.      | Compatibility of Constraints                            | 77 |
| 8.5.        | Errors Concerning Name and Expression Analysis          | 77 |
| 8.5.1.      | Expected Nature of an Entity                            | 77 |
| 8.5.1.1.    | at_first_stm                                            | 78 |
| 8.5.1.2.    | at_others_allowed                                       | 78 |
| 8.5.2.      | Expected Qualities of an Entity                         | 78 |
| 8.5.3.      | Unique Meaning                                          | 79 |
| 8.5.4.      | Usability and Dependencies of Entities                  | 80 |
| 8.5.4.1     | at_discrim_allowed                                      | 81 |
| 8.5.4.2     | at_usable                                               | 81 |
| 8.5.5.      | Completeness of Expressions                             | 82 |
| 8.5.6.      | Syntactic Form of Expressions                           | 83 |
| 9.          | Attribute Types                                         | 84 |
| 9.1.        | at_keys, at_keys1, at_keys2 : tp_key_list               | 84 |
| 9.2.        | at_env, at_env_in, at_env_out : tp_env                  | 84 |
| 9.2.1.      | tp_descr                                                | 85 |
| 9.2.1.1.    | s_designator                                            | 85 |
| 9.2.1.2.    | s_key                                                   | 85 |
| 9.2.1.3.    | s_repr                                                  | 85 |
| 9.2.1.4.    | s_state                                                 | 86 |
| 9.2.1.5.    | s_nature                                                | 87 |
| 9.2.1.6.    | s_den                                                   | 87 |
| 9.2.1.6.1.  | tp_object                                               | 87 |
| 9.2.1.6.2.  | tp_type_den                                             | 88 |
| 9.2.1.6.3.  | tp_constrained                                          | 88 |
| 9.2.1.6.4.  | tp_subpr                                                | 88 |
| 9.2.1.6.5.  | tp_entry                                                | 89 |
| 9.2.1.6.6.  | tp_entry_family                                         | 89 |
| 9.2.1.6.7.  | tp_package                                              | 89 |
| 9.2.1.6.8.  | tp_generic                                              | 89 |
| 9.2.1.6.9.  | tp_enum_literal                                         | 89 |
| 9.2.1.6.10. | tp_dummy_aggr                                           | 89 |
| 9.2.1.6.11. | tp_dummy                                                | 90 |
| 9.2.1.6.12. | tp_operation                                            | 90 |
| 9.2.1.6.13. | tp_void                                                 | 90 |
| 9.2.1.7.    | s_origin                                                | 90 |
| 9.2.1.8.    | s_enclosing                                             | 91 |
| 9.3.        | at_user_reps, at_user_rep : tp_designator_list          | 91 |
| 9.4.        | at_impl_descrs, at_impl_subprs : tp_descr_list          | 91 |
| 9.5.        | at_labels, at_loops_blocks, at_usable : tp_bl_lab_loops | 91 |

|                                                         |     |
|---------------------------------------------------------|-----|
| 9.6. at_allowed1, at_allowed2, at_valid : tp_descr_set  | 91  |
| 9.7. at_context : tp_context                            | 92  |
| 9.8. at_expected : tp_key_void                          | 92  |
| 9.9. at_expected_in, at_expected_out : tp_descr_list    | 93  |
| 9.10. at_init : tp_mixed_init                           | 93  |
| 9.11. at_encl_unit : tp_encl_unit                       | 93  |
| 9.12. at_first_select : tp_select_type                  | 93  |
| 9.13. at_select_state : tp_select_state                 | 93  |
| 9.14. at_first_stm : tp_stm_type                        | 94  |
| APPENDIX                                                | 95  |
| =====                                                   |     |
| A1. Attributes and their Meaning                        | 95  |
| A2. Attributes and their Occurrence in the Rationale    | 99  |
| PART C: THE AG                                          | 101 |
| =====                                                   |     |
| AG0. Introduction                                       | 103 |
| 0.1. Conventions Used in the AG                         | 103 |
| 0.2. Completeness and Parametrization of the AG         | 105 |
| 0.3. Assertions about the Structure of Descriptions     | 106 |
| AG1. Attribute Types                                    | 109 |
| AG2. Constants                                          | 118 |
| 2.1. Designators                                        | 118 |
| 2.2. Constants for Error Values and Abbreviations       | 120 |
| 2.3. Unaccessible Entities                              | 123 |
| 2.3.1. Universal Access                                 | 123 |
| 2.3.2. Universal Integer                                | 123 |
| 2.3.3. Universal Real                                   | 126 |
| 2.3.4. Universal Fixed                                  | 128 |
| 2.3.5. Universal Common Operators                       | 129 |
| 2.3.6. Descriptions of all Unaccessible Entities        | 130 |
| 2.4. Package STANDARD                                   | 131 |
| 2.4.1. Implementation dependent Values                  | 131 |
| 2.4.2. Descriptions of the Entities in Package STANDARD | 131 |
| 2.4.2.1. Boolean                                        | 131 |
| 2.4.2.2. Integer                                        | 133 |
| 2.4.2.3. Float                                          | 136 |
| 2.4.2.4. Character                                      | 139 |
| 2.4.2.5. Package ASCII                                  | 139 |
| 2.4.2.6. Predefined Types and Subtypes                  | 140 |
| 2.4.2.7. Predefined Exceptions                          | 142 |
| 2.4.2.8. Package System                                 | 143 |
| 2.4.3. Description of the Package STANDARD              | 144 |

|                                                |               |     |
|------------------------------------------------|---------------|-----|
| 2.5. Start Environment                         |               | 146 |
| AG3. Terminals and their Attributes            |               | 147 |
| AG4. Nonterminals and their Attributes         |               | 150 |
| AG5. Syntactic Rules and Attribute Calculation |               | 159 |
| 5.2. Lexical Elements                          | r_001         | 159 |
| 5.3. Declarations and Types                    | r_002 - r_080 | 160 |
| 5.4. Names and Expressions                     | r_081 - r_135 | 196 |
| 5.5. Statements                                | r_136 - r_165 | 227 |
| 5.6-7. Subprograms and Packages                | r_166 - r_195 | 243 |
| 5.8. Visibility Rules                          | r_196 - r_197 | 260 |
| 5.9. Tasks                                     | r_198 - r_212 | 261 |
| 5.10. Program Structure                        | r_213 - r_232 | 270 |
| 5.11. Exceptions                               | r_233 - r_236 | 275 |
| 5.12. Generics                                 | r_237 - r_259 | 277 |
| 5.13. Representation Specifications            | r_260 - r_267 | 284 |
| AG6. Semantic Functions                        |               | 291 |
| 6.2 Check and Recovery Functions (cf_)         |               | 291 |
| 6.1 Check Functions (ch_)                      |               | 294 |
| 6.3 Functions (f_)                             |               | 326 |
| AG7. Syntax Cross Reference                    |               | 504 |
| REFERENCES                                     |               | 509 |
| =====                                          |               |     |