AFFIRM

Collected Papers

Roddy W. Erickson, Editor

Version 2.0 - February 19, 1981

Corresponds to ***AFFIRM*** Version 1.21

USC Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, California 90291
(213) 822-1511   ARPANET: AFFIRM@ISIF

The AFFIRM Reference Library

AFFIRM is an experimental interactive system for the specification and verification of abstract data types and programs. It was developed by the Program Verification Project at the USC Information Sciences Institute (ISI) for the Defense Advanced Research Projects Agency. The Reference Library is composed of five documents:

Reference Manual

A detailed discussion of the major concepts behind ***AFFIRM*** presented in terms of the abstract machines forming the system's structure as seen by the user.

Users Guide

A question-and-answer dialogue detailing the whys and wherefores of specifying and proving using ***AFFIRM***.

Type Library

A listing of several abstract data types developed and used by the ISI Program Verification Project. The data type specifications are maintained in machine-readable form as an integral part of the system.

Annotated Transcripts

A series of annotated transcripts displaying ***AFFIRM*** in action, to be used as a sort of workbook along with the Users Guide and Reference Manual.

Collected Papers

A collection of articles authored by members of the ISI Program Verification Project (past and present), as well as an annotated bibliography of recent papers relevant to our work.

Program Verification Project Members

The USC/Information Sciences Institute Program Verification Project is headed by Susan L. Gerhart, with members Roddy W. Erickson, Stanley Lee, Lisa Moses, and David H. Thompson. Past project members include Raymond L. Bates, Ralph L. London, David R. Musser, David G. Taylor, and David S. Wile.

Cover designs by Nelson Lucas.

Special dedication to Affirmed, the only race horse named after a verification system.

This research was supported by the Defense Advanced Research Projects Agency and the Rome Air Defense Command. Views and conclusions are the authors'.

Preface

This volume of the ***AFFIRM*** reference library includes a short bibliography of papers which might be useful for those interested in:

- background on the theory underlying ***AFFIRM***,
- its design,
- specialized application areas, or
- the experience and comments of ***AFFIRM*** users.

A short description is provided for most of these papers.

The *AFFIRM Memos*, short unrefereed notes issued by the Program Verification project, are available upon request[1].

Four papers [Gerhart 80a, Guttag 78a, Guttag 80, Musser 80a] are particularly useful for background and are reprinted in their entirety.

[1] USC Information Sciences Institute, Suite 1100, 4676 Admiralty Way, Marina del Rey, CA 90291 USA

Bibliography

[Bates 80] Bates, R. L.
The Programming of Large Lisp Systems.
Affirm Memo 20, Information Sciences Institute, Program Verification Project, March, 1980.

Explores the problem of building large systems using INTERLISP, particularly the updating and editing of functions and files in an ongoing system. Various problems of such systems are discussed, and a tool that solves some of these problems is shown.

[Berthomieu 80a] Berthomieu, B..
*Proving Progress Properties of Communication Protocols in **AFFIRM**.*
Affirm Memo 35, Information Sciences Institute, Program Verification Project, September, 1980.

Describes a variety of methods which may be used to prove *cycle-freeness*, *boundedness*, and *necessary termination* for state-transition systems described in the axiomatic formalism. Two sample verifications are given.

[Berthomieu 80b] Berthomieu, B.
Selective Repeat Protocol: Axiomatization and Proofs.
Affirm Memo 36, Information Sciences Institute, Program Verification Project, September, 1980.

Using the methods of [Berthomieu 80a], the safety and progress of a communications protocol are proven in ***AFFIRM***. The implications of performing such a proof mechanically are discussed.

[Erickson 80] Erickson, R. W, and D. R. Musser .
The ***AFFIRM*** theorem prover: proof forests and management of large proofs.
In Bibel, W., and Kowalski, R., editors, *Fifth Conference on Automated Deduction*, Lecture Notes on Computer Science, Vol. 87. Springer-Verlag, 1980.
(Also ISI Affirm Memo 13, April, 1980.).

An overview of ***AFFIRM*** 's proof tree mechanism and style of use.

[Erickson 81] R. W. Erickson.
*Enumerated Types and State Machines: Automatic Generation of Rules by **AFFIRM**.*
Affirm Memo 38, Information Sciences Institute, Program Verification Project, February, 1981.

Discussion of the design for a new mechanism. Certain forms of data types can be tedious to define by normal means.

[Flon 79] Flon, L., and J.Misra.
A unified approach to the specification and verification of abstract data types.
In *Proceedings of the Conference on Specification of Reliable Software*, pages 162-169. IEEE Computer Society, April, 1979.

[Gerhart 78]
Gerhart, S. L.
Program verification in the 1980's.
In *Proc. Oregon Report on Computing*, IEEE, 1978.
(Also ISI Research Report 78-71, August 1978.).

Problems, perspectives, and opportunities. A discussion of the trends, problems, and needed breakthroughs in verification.

[Gerhart 79a]
Gerhart, S. L., and D. S. Wile.
Preliminary report on the Delta experiment: specification and verification of a multiple-user file updating module.
In *Proceedings of the Conference on Specification of Reliable Software*, pages 198-211. IEEE Computer Society, April, 1979.

A 1000-line module from an existing program was partially specified and verified using ***AFFIRM.***

[Gerhart 79b]
Gerhart, S. L.
Program validation.
In *Advanced Course on Computing Systems Reliability*. Cambridge University Press, 1979.

A tutorial on testing and proving methods in program verification

[Gerhart 79c]
Gerhart, S. L.
A derivation-oriented proof of the Schorr-Waite marking algorithm.
In F.L. Bauer and M. Broy, editor, *Program Construction, International Summer School*, Lecture Notes in Computer Science, Vol. 69. Springer-Verlag, 1979.

[Gerhart 80a]
Gerhart, S. L., et al.
An overview of ***AFFIRM***: a specification and verification system.
In *Proceedings IFIP 80*, pages 343-348. Australia, October, 1980.

We recommend this as the best overview of our system for beginners. Included in the ***AFFIRM*** Collected Papers.

[Gerhart 80b]
Gerhart, S. L.
Experience with the MITRE Toy Security Kernel.
Affirm Memo 2, Information Sciences Institute, Program Verification Project, January, 1980.

Axioms, partial proof transcript, and a summary of experience from the visit of MITRE personnel Jon Millen and Dave Drake in November, 1979. The Toy Kernel abstracts a machine with processes and blocks of memory, each having a security level. It was proven (in a day's time) that processes gain access to blocks only at their own level. See [Millen 80] for a comparison of this proof with one using HDM.

[Gerhart 80c] Gerhart, S.L.
Induction on Transition Specifications.
Affirm Memo 3, Information Sciences Institute, Program Verification Project, January, 1980.

Outline of the implicit state-vector approach to state transition specifications. State variables are represented as selectors; this is the method we normally use. [Erickson 81] describes a mechanism to facilitate the generation of such specifications.

[Gerhart 80d] Gerhart, S. L.
Connecting Type Instances.
Affirm Memo 27, Information Sciences Institute, Program Verification Project, June, 1980.

A rough proposal on how to connect several instances of a type (such as two "transmitter" nodes in a network). The key problem is expressing (in the ***AFFIRM*** specification method) that the output from one instance is to be input to another. The proposed solution uses demons to move data.

[Gerhart 80e] Gerhart, S. L.
Complete and Recursion Induction in Current ***AFFIRM****.*
Affirm Memo 33, Information Sciences Institute, Program Verification Project, August, 1980.

In proving properties of functions, one needs various kinds of induction. Besides the straightforward data type induction, there are complete induction and recursion induction, both of which require different schemas. Such schemas are presented, justified by carrying out proofs using standard integer induction, and described for other ***AFFIRM*** users.

[Gerhart 80f] Gerhart, S. L.
A Short Blurb on Program Specification Featuring a New Example.
Affirm Memo 34, Information Sciences Institute, Program Verification Project, September, 1980.

A preprint prepared for the Encyclopedia of Computer Science, describing some elementary concepts of specification. An interesting example, an abstraction of a mailing list program, was specified and verified.

[Gerhart 80g] Gerhart, S. L.
Fundamentals concepts of program verification.
In *Proceedings of American Society of Mechanical Engineers, International Computer Technology Conference*, San Francisco, CA., August, 1980.
(Also ISI Affirm Memo 15).

An historical approach to both the theory and practice of program testing and proving.

[Gerhart 81] Gerhart, S. L.
Josephus Circles: An Exercise in Data Structuring.
Affirm Memo 37, Information Sciences Institute, Program Verification Project, January, 1981.

The Josephus Circle problem is interesting because a simple problem-statement and algorithm offer a wide variety of possible data structures. Three implementations--using bit vectors, linked lists, and a special binary tree--were described and verified in ***AFFIRM***. We describe and compare the various implementations and their theories.

[Guttag 75] Guttag, J. V.
The Specification and Application to Programming of Abstract Data Types.
PhD thesis, Department of Computer Science, University of Toronto, October, 1975.

[Guttag 77] Guttag, J. V.
Abstract data types and the development of data structures.
CACM 20:397-404, June, 1977.

[Guttag 78a] Guttag, J. V., E. Horowitz, and D. R. Musser.
Abstract data types and software validation.
CACM 21:1048-1064, December, 1978.
(Also USC Information Sciences Institute RR-76/48, August 1976.).
Included in the ***AFFIRM*** Collected Papers.

[Guttag 78b] Guttag, J. V., and J. J. Horning.
The algebraic specification of abstract data types.
Acta Informatica 10:27-52, 1978.

[Guttag 78c] Guttag, J. V., E. Horowitz, and D. R. Musser.
The design of data type specifications.
In Yeh, R. T., editor, *Current Trends in Programming Methodology*, pages 60-79, Vol. IV Data Structuring. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1978.
(An expanded version of a paper which appeared in Proceedings of the Second International Conference on Software Engineering, October 1976.).

[Guttag 80] Guttag, J. V.
Notes on type abstraction.
IEEE Transactions on Software Engineering SE-6(1):13-23, January, 1980.
Included in the ***AFFIRM*** Collected Papers.

[Lankford 78] Lankford, D. S. and D. R. Musser.
On Semi-deciding First-Order Validity and Invalidity.
(unpublished manuscript.)

[Lee 79] Lee, S., W. deRoever, and S. L. Gerhart.
The evolution of list-copying algorithms.
In *6th Symposium on Principles of Programming Languages*, San Antonio, Texas, January, 1979.

[Lee 80] Lee, S.
*A Numerical Analysis Program Proof in **AFFIRM**.*
Affirm Memo 31, Information Sciences Institute, Program Verification Project, August, 1980.

This experiment involved the transfer of a "hand" correctness proof of the ZEROIN program to ***AFFIRM***. ZEROIN is a widely used, FORTRAN numerical analysis program for approximating a zero of a continuous function; the original (informal) correctness proof is from V. R. Basili and H. D. Mills, *Understanding and Documenting Programs.* (Technical Report TR-884, University of Maryland Computer Science Center, College Park, Maryland, 1980.)

[Loeckx 80a] Loeckx, J.
*Proving Properties of Algorithmic Specifications of Abstract Data Types in **AFFIRM**.*
Affirm Memo 29, Information Sciences Institute, Program Verification Project, July, 1980.

An alternate form [Loeckx 80b] for the specification of abstract data types is briefly surveyed. (All data types include an equivalence relation, and the carrier set is explicitly stated.) The steps needed to render such a specification acceptable to ***AFFIRM*** are detailed, along with the theorems which must be proven in order to certify the validity of the type.

[Loeckx 80b] Loeckx, J.
Algorithmic Specifications of Abstract Data Types.
Technical Report , Universität des Saarlandes (Saarbrücken), 1980.

[London 79] London, R. L.
Program verification.
In P. Wegner, editor, *Research Directions In Software Technology.* MIT Press, 1979.

[Millen 80] Millen, J. and D. L. Drake.
*An Experiment with **AFFIRM** and HDM.*
Technical Report, The MITRE Corporation, Bedford, Mass., December, 1980.

[Musser 77] Musser, D. R.
A data type verification system based on rewrite rules.
In *Proceedings of the Sixth Texas Conference on Computing Systems*, pages .
Austin Texas, November, 1977.

A predecessor to ***AFFIRM***.

[Musser 80a] Musser, D. R.
Abstract data type specification in the ***AFFIRM*** system.
IEEE Transactions on Software Engineering SE-6(1):24-32, January, 1980.

Included in the ***AFFIRM*** Collected Papers.

[Musser 80b] Musser, D. R.
On proving inductive properties of abstract data types.
In *Proceedings of the Seventh ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN, 1980.

If a data type is fully specified, the Knuth-Bendix method can be used to prove theorems, by showing that they are not in conflict with the axioms.

[Schwabe 80] Schwabe, D.
Transport Protocol Specification in ***AFFIRM***.
Affirm Memo 19, Information Sciences Institute, Program Verification Project, March, 1980.

Specifies the service offered by a transport station which allows communication between many users. The service is connection-oriented; users are identified by *port addresses*. Flow control is given by means of *credits* issued by the receiver. Some properties have been proven about the specification; the implementing protocol is not described.

[Schwabe 81a] Schwabe, D.
Formal Specification and Verification of a Connection Establishment Protocol.
(unpublished; forthcoming as an ISI Affirm Memo.)

Describes experience specifying and verifying the '3-way handshake' protocol

[Schwabe 81b] Schwabe, D.
Formal Techniques for Specification and Verification of Protocols.
PhD thesis, University of California at Los Angeles, 1981.
(in progress; forthcoming ISI technical report).

[Stenning 76] Stenning, N. V.
A data transfer protocol.
Computer Networks 1:99-110, 1976.

Describes a 'windowed' protocol; an initial attempt to verify it is in [Thompson 80c]

[Sunshine 79] Sunshine, C. A.
Formal methods for protocol specification and verification.
Computer 12(9):20-27, September, 1979.

[Sunshine 80] Sunshine, C. A.
Axioms for the Alternating Bit Protocol.
Affirm Memo 17, Information Sciences Institute, Program Verification Project, February, 1980.

Presents an early version of both protocol and service specifications in *AFFIRM* for the alternating bit protocol (a simple data transfer protocol between a fixed sender and a fixed receiver over an unreliable medium). Specifications given as axiomatized state transitions.

[Sunshine 81] Sunshine, C. A.
Formal Modelling of Communication Protocols.
Technical Report ISI/RR-81-89, USC Information Sciences Institute, February, 1981.

[Thompson 80a] Thompson, D. H.
*Creating **AFFIRM** Specifications from State Transition Specifications.*
Affirm Memo 4, Information Sciences Institute, Program Verification Project, January, 1980.

A discussion of the pros and cons of explicit *vs.* implicit tuple constructors in the data type specification of a tuple (record) type. See also [Gerhart 80c].

[Thompson 80b] Thompson, D. H.
A Behavioral Axiomatization of the Stenning Data Transfer Protocol.
Affirm Memo 16, Information Sciences Institute, Program Verification Project, June, 1980.

A state transition specification of the Data Transfer Protocol (as defined in [Stenning 76]) in the form necessary for ***AFFIRM***. We informally describe the protocol, and discuss in detail the ***AFFIRM*** formalism. A system invariant is proposed, to (partially) validate the correctness of the specification. An attempt to prove the invariant is described.

[Thompson 80c] Thompson, D. H.
Verification of the Stenning Receiver Process: Progress to Date.
Affirm Memo 14, Information Sciences Institute, Program Verification Project, February, 1980.

Contains the preliminary work necessary to verify Stenning's implementation of the Receiver process, using ***AFFIRM***. (See [Stenning 76], especially Appendix B.) No proof has yet been attempted.

[Thompson 80d] Thompson, D. H., Bates, R. L., and Erickson, R. W.
***AFFIRM**-120: Changes since v-104.*
Affirm Memo 32, Information Sciences Institute, Program Verification Project, October, 1980.

A detailed description of the enhancements and modifications made to ***AFFIRM*** from late May, 1980 through early October, 1980.

[Thompson 81a] Thompson, D. H., and R. W. Erickson.
*Documentation of the Proofs for the **AFFIRM**-Protocol Paper.*
Affirm Memo 39, Information Sciences Institute, Program Verification Project, January, 1981.

A compilation of the transcripts of the proof of the theorems and lemmas referenced in [Thompson 81b]

[Thompson 81b] Thompson, D. H., C. A. Sunshine, R. W. Erickson, S. L. Gerhart, and D. Schwabe.
Specification and Verification of Communication Protocols in ***AFFIRM*** *using State Transition Models*.
Technical Report ISI/RR-81-88, USC Information Sciences Institute, February, 1981.
(Also submitted for publication).

[Thompson 81c] Thompson, D. H., S. L. Gerhart, R. W. Erickson, S. Lee, and R. L. Bates, eds.
The ***AFFIRM*** *Reference Library*.
USC Information Sciences Institute, 1981.
5 vols: Reference Manual, User's Guide, Type Library, Annotated Transcripts, and Collected Papers; 500 pages.

[Wile 79] Wile, D. S.
POPART: Producer of Parsers and Related Tools.
(Information Sciences Institute Report, in preparation.)

[Wing 80] Wing, J. M.
Experience with Two Examples: A Household Budget and Graphs.
Affirm Memo 30, Information Sciences Institute, Program Verification Project, August, 1980.

A user evaluates ***AFFIRM*** as an interactive software tool for specification during program development. Example specifications were built for graphs and a simple household budget database. A number of extensions and improvements to the system are suggested.