

- 13- The ability to coerce any data type into any other, where the coercion is logical. In particular the executable structure code can be coerced into type string for display or editing. The inverse of this coercion is compilation.
- 14- A distinction is made between abstract and concrete manifestations of encoding and structure. It is postulated that lack of this distinction results in some fuzzy notions in very high level languages.
- 15- The notion of conserved attributes is reviewed and the concept of literals is generalized under the term Reflexivity.
- 16- The notion of assignment is examined in terms of source attributes propagated to the sink. A philosophy is expressed for assignments involving bulky objects.
- 17- The motivations and techniques of language design are examined. The principles of notation synthesis are examined in some detail and the possibility of automating macro and notation synthesis is proposed (3,6).
- 18- The heuristics employed in optimizing paraphrase generation and operator attribute derivations are reviewed.
- 19- The psychological and pragmatic motivations for the choice of data structures such as strings, integers and vectors are examined.
- 20- The problem of optimization in the extension framework is examined, in particular, efficiency is postulated to be related to a critical point of extension. In addition, optimizations appropriate to each layer of language extension are discussed.
- 21- The role of environment inquiries and compumetric experiments is examined in regard to program optimization.
- 22- A technique is examined which permits crude definition of operators such as + and < in terms of low level semantics (5).
- 23- Reflections of formal definitional facilities for host CPU data structures, operation code semantics and machine architecture of miscellaneous sorts. It is hoped that such information is useful in generation of machine dependent optimizations.
- 24- The notion of phantom problems is defined and it is shown that they are the major unnecessary complexity factor encountered in programming.

Documentation:

Existing Documentation and Cited References:

- (1) Harrison, M. C. An Introduction to the BALM Programming Language. New York University - Courant Institute Computer Sci. Dept. 1972.
- (2) Schwartz, J. T. On Programming. New York University - Computer Science Dept. February 1973.
- (3) Pfeffer, L. Basic Thoughts on Optimization. New York University - Courant Institute, Computer Sci. Dept. 5-1972.

- (4) Pfeffer, L. Preliminary Specifications for the DAHLIA Notation and an Anthology of Optimizing Paraphrases. New York University - Courant Institute, Computer Sci. Dept. 1973.
- (5) Pfeffer, L. Reflections on BALM. BALM newsletter No. 22, New York University - Courant Institute, Comp. Sci. Dept. 2-73.

Potential Documentation and Cited References:

- (6) Pfeffer, L. Confluents, De-Compilers and Notation Generators, or Applied Artificial Intelligence in Programming Languages. New York University - Courant Institute, Comp. Sci. Dept. (est. June 1973).

Name of Contracts for Future Reference:

Lawrence Pfeffer, 152 Parkside Avenue, Brooklyn NY 11226

 ***** PERMANENT AND TRANSIENT DATA STRUCTURES AND *****
 ***** ASSOCIATED OPTIMIZATION TECHNIQUES *****

Name of Project: Permanent and Transient Data Structures and Associated Optimization Techniques

Principal Investigator: Lawrence Pfeffer

Address: 152 Parkside Avenue, Brooklyn, NY 11226

Description of Project:

Objectives:

- (1) Elimination of the gross notational distinction made between structures resident in various memory hierarchies.
- (2) Delineation of responsibility for algorithm elaboration between language processors and operating systems.
- (3) Investigation of optimizations applicable to (or mandated by) bulky structures, at both levels of responsibility.

Starting Date: 8/72

Completion Date: 3/73

Current Status:

The results of the research are being compiled with a view toward containment in a research paper.

Computers Being Used: IBM S/370

(Most of the research has been theoretical, however it is based on a generalized query processor implemented approximately two years ago.)

Languages Being Used: MAGUS/360 (1)

(A systems implementation language, which is in this case an anthology of approximately 100 BAL macros and logic modules.)

Methodology or General Approach:

A dictionary is developed in SETL (2) for unified treatment of data structures resident in core storage and structures contained on files. The sole distinction between the two structures is shown to be that of life-time and the two structures are named transient and permanent, respectively. Data structure temporality is declarable. Since the gross distinction between the two structures is removed, large transient structures are implicitly migrated between memory hierarchy levels and permanent structures can be subjected to all the operations defined for less distinguished data structures. Accordingly, generalized data management system functions such as querying and updating seize to be special processes and can be optimized by techniques developed for algorithmic languages. A set of optimizations is developed which minimize the search space of structures. Other optimizations based on data compression and access postponement are also investigated. Finally optimizations applicable to parallel processors such as the STARAN are discussed. This optimization is especially significant because it is the first significant application of parallelism in a widely applicable problem domain. All optimizations are applicable to both transient and permanent data structures. The notion of (simple) job control statements is discussed and a maze of dedicated permanent structures called system catalogues is investigated. Optimizations applicable to bulky structures are examined from the point of view of the operating system. The main techniques discussed are: resource load leveling, early resource release, late resource allocation, disc arm seek and rotational delay minimization. Symbolic access to fields of both types of structures is based on templates, which define logical field relations. Structures may be passed in function calls without regard to temporality and a technique called aliasing is presented for minimization of needless object copying. Templates are passed in function calls and minor template incompatibilities between passed and expected structures are automatically adjusted at the interface in the form of a suitable coercion. This feature is significant on its own, since it eliminates many bugs occurring at the module interface, A set of entities called fussy readers and writers is examined, that is entities which place severe constraints on the cosmetics and encoding of information presented to them. Some interesting language dictionaries are devised in the course of uniting the notions of structures-in-core and structures-on-mass-memory. One dictionary is the existence granter / killer / existence checker triplet. Another dictionary permits reference to structures by approximate arguments. This notion permits significant optimizations with regard to limiting the search space of large structures. It also provides the opportunity to rethink the interpretation to be assigned to function arguments

and structure locators. The notion of infinity emerges and is permitted to become explicit in controlled circumstances. This also provides the opportunity to rethink some old notions and provides a welcome intellectual shock. Many of the ideas and the basic approach are aided by the work of Prof. J. T. Schwartz (2).

Documentation:

Existing Documentation and Cited References:

- (1) L. Pfeffer. MAGUS: A Systems Implementation Tool for the IBM S/360-370 Family of Computers. National Health and Welfare Ret. Assn. - Systems Division, 360 Park Ave South, New York, NY. March 1970.
- (2) J. T. Schwartz. Preliminary reflections on the use of SETL in a Data-Base Context. New York University - Courant Institute, SETL Newsletter 90, Dec. 1972.
- (3) L. Pfeffer. QL/1 - An On-Line Query Language. Morgan Guaranty Trust, Methods and Systems Div. 30 W. Broadway, New York, NY. June 1971.

Potential Documentation:

- (4) L. Pfeffer. Permanent and Transient Structures and Associated Optimization Techniques. New York University - Courant Institute, Comp. Sci. Dept. (To be issued as a SETL Newsletter sometime before June.)

Name of Contacts for Future References:

Lawrence Pfeffer, 152 Parkside Avenue, Brooklyn NY 11226.

*
* RAMBAM - A NOTATION AND METHODOLOGY FOR MAN-MACHINE
* DIALOGUE SPECIFICATION
*

Name of Project: RAMBAM - A Notation and Methodology for Man-Machine Dialogue Specification

Principal Investigator: Lawrence Pfeffer

Address: 152 Parkside Avenue, Brooklyn, NY. 11226

Description of Project:

Objectives:

- (1) Definition of a language suitable for dialogue specification.
- (2) Separation of processing and dialogue logic fabrication to permit division of labor, human engineering of dialogues and relative data encoding independence.

Starting Date: 6/71

Completion Date: 3/72

Current Status:

This research has been completed some time ago, however reporting its findings is contemplated only now.