
The application of Burst processing to the problem of spectral decomposition of speech provides a viable alternative to conventional speech analyzers. A specific Burst implementation is presented.


This paper presents an algebraic definition of Knuthian semantic systems (K-systems or attribute grammars) with both synthesized and inherited attributes. The approach is based on the initial algebra semantics principle formulated by Goguen, Thatcher, Wagner, and Wright. Given a K-system semantic definition for a context-free grammar G, it shows how to construct a many-sorted algebra K such that the semantic mapping from G-derivation trees into their “meaning” is the unique homomorphism from the initial algebra T(1) of derivation trees into K. The practical implications of the algebraic definition of K-systems are discussed, and the combined use of Knuth’s original formulation and the algebraic approach for the development of semantic definitions is advocated.


The Fortran program ILLOD—(AND/OR-B) derives optimal combinational networks with AND and OR gates for one output function or a set of output functions. The algorithm is based on the branch-and-bound method. The optimality of networks is defined as C = AR + BI where R and I are the numbers of gates and connections, respectively. Coefficients A and B can be set by users. Incompletely specified functions also can be processed.


ILLIP-2 is an improved version of the program ILLIP. This report describes new features added to the ILLIP-2 and how the ILLIP is modified. Comparisons of some computational results between the ILLIP-2 and the ILLIP are shown.


The network transduction programs, including NETTRA-PG1, -P1, -P2, -G1, -G2, -G3, -G4, -E1, and -E2 are combined as a large program system named the NETTRA system (NETWork TRAnsduction system). It can design near-optimal, multiple-output, multi-level and loop-free NOR(NAND) networks under fan-in/fan-out restrictions and/or level restriction. Given function(s) may be completely or incompletely specified, and both complemented and uncomplemented external variables are permitted as inputs. The user can specify the control consequence (the types of the initial network methods and the types and the order of the transduction procedures to be applied) to solve his
problem. Besides, four control sequences are provided for users who are not interested in the details of how to specify the control sequence. Also supplied are facilities for treating unfinished jobs due to the expiration of computation time.


This is the program manual for the NETTRA system, which can design near-optimal, multi-level and loop-free NOR (NAND) networks under fan-in/fan-out restriction and/or level restriction. Given functions can be completely incompletely specified, and both complemented and unimplemented external variables are permitted as inputs. The user can specify the control sequence (the types of the initial network methods and the types, and the order of the transaction procedures to be applied) to solve his problem. Besides, four control sequences are provided for the users who are not interested in the details of how to specify the control sequence. Also supplied are facilities for treating unfinished jobs due to the expiration of computation time.

R77-363—Zeckowitz, Marvin V., and H. J. Larsen, "Implementation of a Capability-Based Data Abstraction" (33 pp., University of Maryland, College Park, Maryland)

A significant feature in programming language design is an appropriate data definitional facility. Criteria now recognized as important are the concepts of information hiding and data abstraction. The problem, however, is to embed these ideas into languages. Although including these ideas has often led to the design of a new language, this is not always necessary. Such facilities may be added to languages like PL/1 or Pascal. This report discusses the inclusion of the facilities within one such PL/1 compiler. While the resulting system does not have the optimal set of protection features, it does have several advantages: the base language is known to a large class of programmers, there are many such compilers already written, and the system achieves almost as much protection as is needed.

R77-364—De Millo, Richard, D. Dobkin, and R. J. Lipton, "Even Data Bases that Lie Can Be Compromised" (10 pp., Georgia Institute of Technology, Atlanta, Georgia)

Users can compromise data bases by asking a series of questions, even when the data bases are allowed to lie.

R77-365—Airapetian, Artashes N., and J. F. McDonald, "A Subscripted D-Algorithm for Multiple Simultaneous Test Generation for Combinational Circuits" (60 pp., Polytechnic Institute, Erevan, USSR/Ruths, Troy, New York, respectively)

The problem considered in this paper is that of test set generation for an arbitrary gate in a combinational circuit which is constructed from AND's, NAND's, OR's, NOR's, and NOT's. A new algorithm is given (AALG-11), which can generate many tests simultaneously for multiple input gates in a combinational circuit. For Boolean tree type circuits (and for Boolean graph-circuit in the most favorable circumstances) the new algorithm generates tests for all failures of a gate to be checked simultaneously. It uses a new type of D-cube which we call a subscripted D-cube, and an associated intersection operation. These can be used to compute a kind of receive for producing the appropriate test inputs. The key idea is that once a path is sensitized it is suitable for checking many singular cubes of a gate provided no conflicts arise during the various phases of the algorithm. One of the goals of the algorithm is the minimization of these conflicts.

R77-366—Bender, Edward A., and J. T. Butler, "Asymptotic Approximations For the Number of Fanout-Free Functions" (14 pp., Northwestern University, Evans- ton, Illinois)

Expressions are derived for the approximate number of functions realized by various n-variable fanout-free networks. Six recently studied networks are considered. It is shown that the relative number of functions realized by two networks for small and large n is quite different in certain cases.

R77-367—Cavallo, R., R. Potenza, and F. Riggi, "Automatic Analysis of Spectra of Charged Particles Produced in Nuclear Reactions" (14 pp., Instituto di Fisica Nucleare dell'Universita, Corso, Italy)

This paper describes a method of analyzing charged particles spectra obtained in nuclear reactions studies at low energy. It is composed of the following steps: 1) peaks identification, 2) background subtraction, 3) separation of partially overlapped peaks, and 4) determination of cross-sections and errors. The method allows results to be rapidly obtained with a high precision. A test of the program by means of a Monte Carlo method, in case of overlapping peaks, is given.

R77-368—Barnard, David (ed.), "An Annotated Bibliography on Computer Program Engineering" (17 pp., University of Toronto, Canada)

An outgrowth of a graduate course on computer program engineering, this bibliography is an attempt to provide some guidance to the literature in this area. References annotated by students in the course are provided, along with a cross reference list by topic area.

R77-369—Greenspan, Sol J., and J. J. Horning (eds.), "Programming Methodology: An Annotated Bibliography for IFIP Working Group 2.3" (64 pp., University of Toronto, Canada)

This bibliography contains entries for nearly 200 books, articles and papers representing the work and interests of IFIP Working Group 2.3 on Programming Methodology. This edition is a first, and very preliminary, version of what should become a useful guide to an important core of the programming methodology literature.

R77-370—Nanya, Takashi, and Y. Tohma, "Universal Multicode STT State Assignments for Asynchronous Sequential Machines" (23 pp., Nippon Electric Company, Ltd., Kawasaki City, Japan)

This paper presents a new systematic procedure for constructing universal STT state assignments for asynchronous sequential machines. It further shows that the procedure can also be applied to improve the best known bounds for (2,1)-separating systems. The feature of the procedure lies in that two binary code vectors are assigned to each internal state. In the resulting universal STT assignments, the number of state variables required is given by 2m' + n where m = \lceil \log_2 n \rceil, n being the number of states, and [x] being the least integer \geq x.

Furthermore, the resulting (2,1)-separating systems require 2m state variables.

R77-371—Cerny, E., "Unique and Identity Solutions" (12 pp., Concordia University, Montreal, Canada)

As the range of applications of Boolean equations in logic design increases, there is a need for efficient procedures which would either solve these equations or at least provide information about the properties of the solutions. In this respect, two algebraic methods are proposed, one for determining whether an equation has a unique solution, the other for finding identity constants). Both methods are easily implementable on a digital computer—for instance, using cubical array structures and operations.

R77-372—Cerny, E., D. Mange, and E. Sanchez, "Synthesis of Minimal Demultiplexer, Multiplexer, and Binary-Decision Trees" (49 pp., Concordia University, Montreal, Canada)

The concept of binary-decision trees is extended to include multiple output Boolean functions. A method based on the properties of standard sets and the methodology of characteristic functions is then developed for the minimization of trees, realizing multiple output, incompletely specified functions. Furthermore, physical realizations of the decision trees using decision programs, demultiplexer, multiplexer and contact networks are developed, showing their mutual transformation rules.
R77-373—Stankovic, John, and A. van Dam, “The Distributed Processing Work-
shop” (108 pp., Brown University, Providence, Rhode Island)

This report contains a session-by-session edited transcription of the second dis-
tributed processing workshop held at Brown University, August 3-5, 1977.

R77-374—Murthy, K. R. Srinivasa, “Transformation from Boolean Algebra to Arith-
etic or to Ordinary Algebra” (6 pp., Department of Space, Bangalore, India)

This paper presents a method of transform-
ation for Boolean algebra to ordinary
algebra. This transformation would enable
any computer with only an arithmetic operation facility to deal with Boolean
algebraic expressions.

R77-375—Gimson, R. B., “Data Represen-
tation by Program Transmission” (25 pp.,
University of Newcastle upon Tyne, England)

This paper describes an approach to
data representation, based on transform-
ation applied to the source language form
of a program. Such transformations provide a concise machine-manipulable
way of writing data representations, which
may be checked for correctness and stored
in libraries for use during program develop-
ment. Some requirements for expressing
such transformations are demonstrated with
two example representations, as well as
each of a notation that satisfies the require-
ments. An example of how the transforma-
tion approach may be applied during program development, and the charac-
teristics of the approach, are included.

R77-376—Graham, Scott G. (ed.), “Topics
in Queueing Network Modeling” (133 pp.,
University of Toronto, Canada)

Queueing network models are finding
growing acceptance as useful, cost-effective
tools for evaluating computer system per-
f ormance. This report is a collection of
related papers on queueing network model-
ing by members of Project SAM in the
Computer Systems Research Group. Of
the seven papers included, five have been
e xternally-refereed and will be published in
conference proceedings.

R77-377—Graham, Scott G., “A Study of
Program and Memory Policy Behavior”
(244 pp., University of Toronto, Canada)
Memory management policies are surveyed,
in particular the Page Fault Frequency
and Denning Working Set variable partit-
don policies. PPF is subject to both
anomalous and gap behaviors, not exhibited
by DWS. These behaviors make PPF
difficult to control: additional experiments
on adjusting the memory policy parameters
confirm this observation. Using trace
tape information, certain features of the
lifetime function and space-time cost for
DWS and PPF are related. This relation-
ship is developed further in a queueing
network model, in an investigation of the
lifetime knee heuristic for optimal system throughput.

R77-378—Tsichritzis, D., “Research Direc-
tions in Data Base Management Sys-
tems” (131 pp., University of Toronto,
Canada)

This is a collection of short papers outlining some of the research work carried
out in the data base group during mainly
the year 1976. It is just a collection of the
following different ideas: 1) research direc-
tions in DBMS, 2) data base constraints,
3) an example of ANSI/SPARC archi-
tecture, 4) the implementation of LSI, 5) a
front end DBMS, 6) user performance
measures in DBMS, and 7) human factors experiments in DBMS.

R77-379—Thompson, David H., “The De-
sign and Implementation of an Advanced
LALR Parse Table Constructor” (109 pp.,
University of Toronto, Canada)

This report describes the design and
implementation of an LALR(1) parse table
constructor and a parser utilizing the
tables. The constructor is noteworthy
because it allows semantic action symbols
to be embedded at near-arbitrary points
of each production, and because regular
right parts are allowed in productions.
Included are: the theory of L(1) pars
parse table construction using Aho and Ullman’s model involving pairs of functions
(f,g); DeRemer’s incremental approach (LR(0),
SLR(1), and LALR(1)), still in terms of the
(f,g) model; the pieces of the systems in
detail, including the parser, grammar nor-
malization, grammar analysis, table con-
struction, and table optimization; and a
discussion of the theory behind the embed-
ding of semantic action symbols. The work
is related to that of Lewis and S terns in
terms of their “Derived Symbol Polish
grammal.

R77-380—Kohonen, Teuvo, and H. Rittlin-
en, “A Preliminary Proposal for
the Recognition of Two-Dimensional
Patterns Shown in Different Perspec-
tives” (9 pp., Helsinki University of Tech-
ology, Finland)

This paper addresses the recognition of
perspective images of two-dimensional
patterns. A transformation function of
the perspective image is shown to be in-
variant with respect to different relative
locations of the original pattern, whereby
recognition can be based on the matching
of such transformation functions.

R77-381—Parker, Alice, C., A. Nagle, and
J. Gault, “Structure and Function of a
General Processor Input/Output Pro-
cessor” (pp., Carnegie-Mellon Univer-
sity, Pittsburgh, Pennsylvania)

This paper describes a processor archi-
tecture designed specifically to perform
input/output and interfacing functions for
any central-processor-peripheral config-
uration. It is justified on the basis of
functional I/O requirements, which are
discussed in detail. This processor is micro-
programmable with a writeable control
store, allowing dynamic configuration of
the processor for different input/output
and interfacing applications. Underlying
the microcontrol is a ROM-resident nano-
program which performs the complex
timing, handshaking, and bookkeeping
control tasks. The processor architecture
is modular and bus oriented.

R77-382—Trivedi, Krishor S., “PWS: An
Optimal Variable Space Preparing Algo-
rithm” (18 pp., Duke University, Durham,
North Carolina)

This paper defines a new variable space
preparing algorithm, PWS, which incurs
zero page faults, fetches the same number
of pages, and requires the same amount
of memory space as Denning’s WS algorithm.

R77-383—Trivedi, Krishor S., “An Analy-
gsis of Preparing” (40 pp., Duke University,
Durham, North Carolina)

Preparing is advocated as a technique
to reduce the excessive page traffic due to
the changes in the phases of program
execution. Common preparing techniques
are surveyed. The phase transition behavior
cannot be adequately predicted based either
on the spatial contiguity or on the
observation of the past behavior. Preparing
advice generated by the programmer or
the compiler is presented as a technique
for the prediction of the phase transition
behavior. To simplify the generation of
the preparing advice, the processes of the
extraction of the phase transition behavior
and the scheduling of the page transfers
are decoupled. This, in turn, dictates the
need for controlled preparing, which is
subsequently discussed.

R77-384—Takiyama, Ryuzo, “A Relation-
ship Between the Cascaded Network
and the Network of Threshold
Logic Units for Pattern Recognition”
(11 pp., Kyushu Institute of Design,
Fukuoka, Japan)

The cascaded threshold logic unit (TLU)
network and the two-layer TLU network
are two of fundamental TLU networks.
There always exists a two-layer network
(with an appropriate number of units in
the first layer) which realizes the same
discriminant function as that of any given
cascaded network; furthermore, parameters
of the two-layer network are determined
directly only by that of the cascaded network.

R77-385—Blauw, G. A., A. Dunworth,
W. J. M. Geerdink, and C. A. Vissers, “The
Use of APL for Design and Description of
Digital Systems” (31 pp., Twente Univer-
sity of Technology, Enschede, The Nether-
lands)

This paper describes the philosophy of
the Digital Technique sub-department of
the Twente University of Technology
with respect to the design of digital systems,
including broad outlines of the use of APL
in a number of major research areas.

Order by R-Number. Use the Repository order form on p. 104.