A LISP II PROGRAMMING SYSTEM

Proposal No. 64-12

30 December 1964

Submitted to:

System Development Corporation
2500 Colorado Avenue
Santa Monica, California

Attention: Mr. William Barancik

TABLE OF CONTENTS

I INTRODUCTION AND SUMMARY

This document presents a proposal from Information International, Inc. to System Development Corporation for the design, development, implementation and documentation of a LISP II system. The proposed project would be carried out jointly by I.I.I., SDC and Project MAC personnel, with I.I.I. providing its services on a cost-plus-fixed-fee basis. I.I.I. has had considerable experience in the development and application of LISP, and it can be anticipated that the collaboration between SDC, Project MAC and I.I.I. will be a fruitful one. Professor John McCarthy of Stanford University has endorsed this project, and he will provide additional advice and guidance.

LISP II will combine the symbol manipulating capability of LISP 1.5 with the linguistic convenience and power of ALGOL 60; LISP II is thus a natural outgrowth and merger of these two languages. The source language of LISP II will be a slightly modified form of ALGOL; the data types and the structure of compiled programs will be drawn mainly from LISP. Unlike LISP 1.5, LISP II will perform arithmetic in an efficient fashion, so that LISP II will be suitable for both symbolic and numerical calculations. LISP II will first be implemented on the AN/FSQ-32 computer and later on the GE 635 computer, under time-sharing; both the internal structure of the LISP II system and the communication facilities with the user will be oriented towards effective operation in a time-shared environment.

Part II of this proposal describes the scope and work plan of the project. Section A, Statement of Work, specifies the items to be delivered; and Section B, Project Tasks and Schedule, specifies the steps of the work plan and their scheduled durations.

Part III presents a technical discussion of LISP II. Section A contains a description of the source language which is used as a basis for specifying the properties of the language. Sections B and C describe the treatment of data within LISP II and the translating techniques to be used.

Part IV consists of a description of I.I.I. and its personnel. The first section describes the company; a second section contains resumes of key I.I.I. personnel.

Part V contains an estimate of costs.

II PROJECT SCOPE AND ORGANIZATION

A. Statement of Work

LISP II will be produced as a joint effort of I.I.I., SDC, and Project MAC. I.I.I. will be responsible to SDC for the actual delivery of the system, and will provide technical personnel to participate in the project. Project MAC will provide technical direction of the project, and SDC will provide technical assistance. If any deviations from the project plan or schedule are necessary, these will be jointly agreed upon by Project MAC and I.I.I. in writing, and approved by SDC. I.I.I. will provide a total of approximately 48 man-months of direct technical labor; the choice of technical personnel involved will be agreed upon by all three parties. SDC will provide approximately 36 man-months of technical assistance, and Project MAC will provide approximately 11 man-months of technical direction. These manning levels are based on the 18-month span of the project plan, although the cost proposal is based on the 10 months of the proposed initial contract. The following items will be delivered:

(1) Pre-programming specifications of the Syntax Translator, the Syntax Translator Generator, the Source Language, the Intermediate Language, the Internal Conventions, and the Compiler for LISP II.

(2) An operative LISP II system for the Q-32 computer, in conformance with the technical specifications set forth in Part III of this proposal.

(3) An operative LISP II system for the GE 635 computer under time-sharing, in conformance with the technical specifications set

forth in Part III of this proposal.

(4) A LISP II Primer providing the user with a simple and pedagogically oriented explanation of how to write programs in the source language.

(5) A LISP II Reference Manual providing a systematic and complete description of the language and all its features, and specifying any parts of the language that are not machine independent.

(6) A LISP II System Manual describing the implementation techniques used on both the Q-32 and the GE 635, the internal conventions, how new data types are added, and how to modify the syntax translator using its generator. Machine mobility will also be discussed.

(7) Intermediate documentation as required, describing progress to date and future plans.

B. Project Tasks and Schedule

In order to specify the work plan, the project has been divided into a set of tasks. The schedule for these tasks is shown in Figure 2-1. During the progress of the project, changes in the schedule, the task definitions, or both may become necessary; any such changes will be agreed upon by I.I.I. and Project MAC in writing. Documentation will be performed concurrently with other tasks. The tasks and their descriptions are:

(1) Task 1: Specification of the Source Language. A rigorous definition of the syntax and semantics of the source language will be developed and documented.

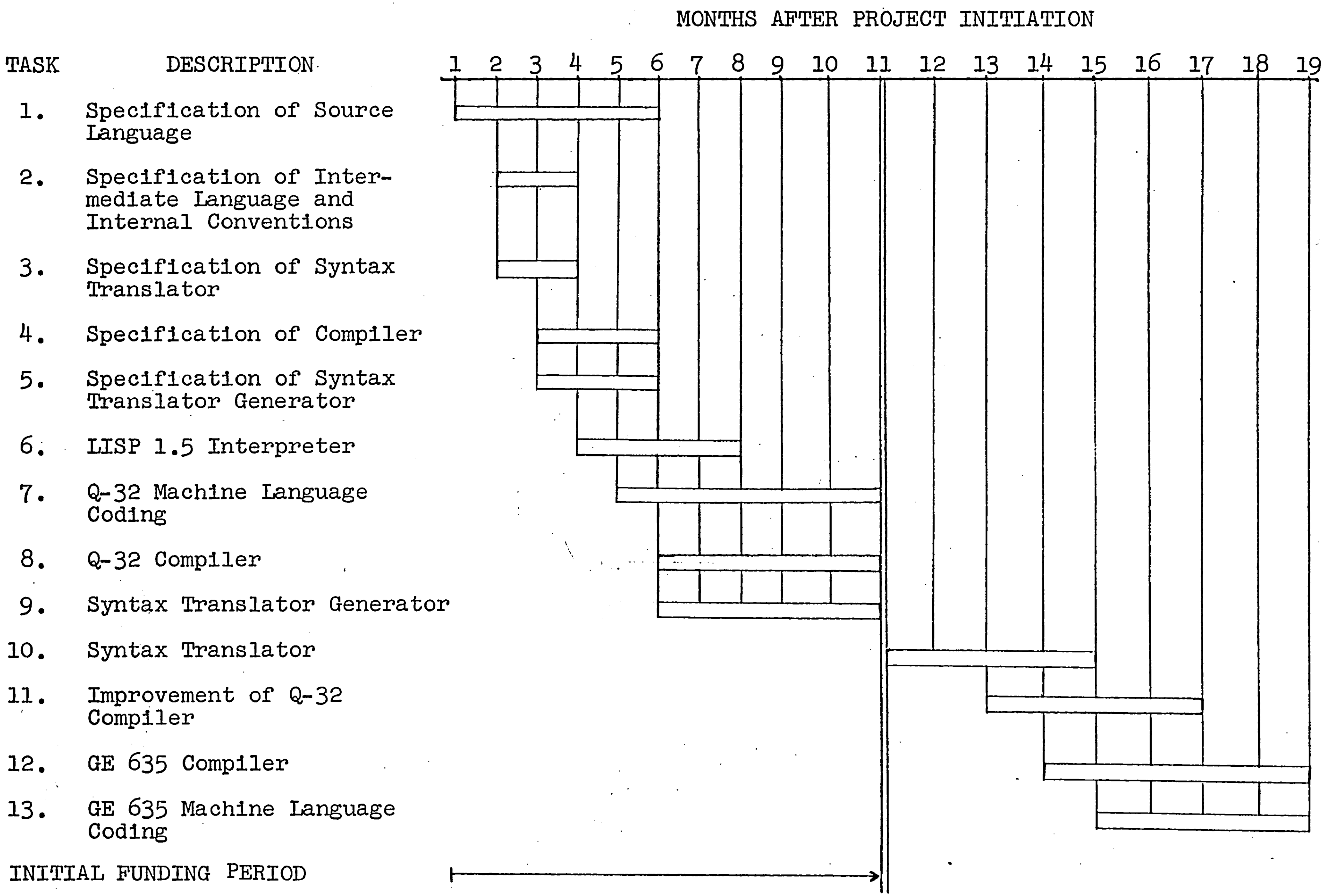

Figure 2-1. PROJECT SCHEDULE

(2) Task 2: Specification of the Intermediate Language and Internal Conventions. The syntax and semantics of the intermediate language will be specified jointly with the internal conventions; these specifications will be documented.

(3) Task 3: Specification of the Syntax Translator. The syntax translator will be specified and documented.

(4) Task 4: Specification of the Compiler. The compiler will be specified and documented.

(5) Task 5: Specification of the Syntax Translator Generator. The syntax translator generator will be specified and documented.

(6) Task 6: LISP 1.5 Interpreter. An interpreter for a subset of the intermediate language will be programmed within LISP 1.5, using the CTSS system on the Project MAC IBM 7094 and the SDC Time Sharing system on the Q-32 computer.

(7) Task 7: Q-32 Machine Language Coding. The basic machine-language programs required to make LISP II operative on the Q-32 will be written. These programs include input-output, the basic LISP functions, the reclaimer, the assembler, and the linking loader.

(8) Task 8: Q-32 Compiler. A compiler for generating Q-32 machine language code from intermediate language programs will be written in the intermediate language subset and interpreted, using the LISP 1.5 interpreter, thus producing a compiler operative on the Q-32.

(9) Task 9: Syntax Translator Generator. The syntax translator generator will be programmed in the intermediate language.

(10) Task 10: Syntax Translator. The syntax translator will be programmed in the language of the syntax translator generator.

(11) Task 11: Improvement of Q-32 Compiler. The Q-32 compiler will be rewritten in the LI [illegible] II source language and improved.

(12) Task 12: GE 635 Compile [illegible] he compiler for the GE 635 that takes intermediate langua [illegible] code to GE 635 machine language will be written in source la [illegible] ge and executed on the Q-32 to produce a GE 635 machine langu [illegible] compiler.

(13) Task 13: GE 635 Machine La [illegible] ge Coding. The basic functions described in Task (8) [illegible] l be programmed in machine language for the GE 635.

III TECHNICAL DESCRIPTION

A. <u>The LISP II Source Language</u>

The capabilities of LISP II and the requirements for its implementation are best described in terms of its source language. In this section a general specification of the source language will be developed; the following two sections will consider the implications of this specification in terms of data-handling procedures and in terms of translation techniques.

The LISP II Source Language will closely resemble ALGOL 60, and will in fact be upward compatible with ALGOL 60 so that any ALGOL 60 program will be a legitimate LISP II program. Consequently, LISP II will have the full algebraic capability of ALGOL 60, and will be able to operate on Boolean, integer and real data with no loss of efficiency. Moreover, in LISP II it will be possible to add new data types within the framework of the language and to have the arithmetic operations treated correctly for these types. For example, it would be possible to define complex numbers as a new data type, and to have complex multiplication performed whenever the multiplication sign appears between two complex quantities.

Experience has shown that convenient array-handling facilities are essential to an algebraic computer language, and LISP II will incorporate significant advances over ALGOL 60 in its array-handling features. Array storage will be assigned dynamically, and the source language will include facilities for creating or destroying arrays at any time. Furthermore, it will be possible to treat entire arrays in the same manner as single elements, and functions on arrays will be definable directly.

S-expressions will also constitute a data type, replacing the ALGOL data type "string". The elementary S-expressions can be identifiers, character strings, numbers, arrays, or any other data type definable within the system. The manipulation of S-expressions will be greatly facilitated through the inclusion of string transformation procedures based on the COMIT rule. These procedures will permit the programmer to match a list of constituents with a format and then rearrange the elements as specified by a different format. It will be possible to call LISP II procedures from within a string transformation so that this procedure will be far more powerful than the standard COMIT rule.

The ALGOL "call by name" will be replaced by three new types of call: call by location, functional call, and formal call. The call by location causes the appropriate dummy variable of the called procedure to be identified with the name of the corresponding variable in the call; the call by location is almost but not entirely identical to the ALGOL call by name. An appropriate mechanism will be included to preserve compatibility. The functional call provides a mechanism whereby a function definition can be provided as an argument in a calling sequence; this facility has no analogue in ALGOL. The formal call is essentially a functional call where the function specified in the call takes no arguments, and can therefore be specified more concisely.

There has been a trend both in ALGOL and in LISP, as the languages have evolved, towards more and more use of declarations. Declarations are necessary in order to define data types and resolve certain other ambiguities; they are also used under some circumstances to enable a program to operate more efficiently. LISP II will require more types of declarations than either LISP 1.5 or ALGOL 60; however, in all cases where a declaration is expected, a standard assumption will be made in the absence of a declaration. Thus, the programmer can ignore questions of efficiency if he chooses, and will need to declare only enough information to prevent compilation of an erroneous program. The language will be so constructed that any syntactically correct program will remain syntactically correct when all declarations are removed from it.

B. Storage and Data Conventions

The requirements for time-sharing operation are an essential consideration in establishing the storage conventions for LISP II, since it is desirable both that no program should use more storage than it actually needs, and that self-modifying procedures be avoided. Conservation of storage is essential since the storage freed by one user can be utilized by another; avoidance of self-modifying procedures is essential because it permits one copy of a program to be called by several different users.

Each user will at any time be assigned a certain amount of storage; when he exhausts it he can either call the reclaimer program that will recover his dead space or he can request more storage from the system. The decision as to which alternative to take can be made at run time, and can be based on information about the entire state of the system. Memory will be divided into public and private areas. The public areas will contain system programming and library routines; the private areas will contain users' own procedures and data. Unlike LISP 1.5, it will be possible to intermix these randomly. When a user calls a public or library procedure, the required storage will be taken from his own space; thus several users can call the same procedure, and in a multi-processing system they can even all use it at once.

The basic type of data will be lists and blocks; lists can lead into blocks and blocks can contain pointers to lists. Both lists and blocks will be relocatable, and they will be intermixed randomly. Unused storage space can be recovered either through the reclaimer or by restoring it directly; thus it will be possible to have S-expressions and symmetric lists, say, in the same storage area, and to reclaim one and restore the other. The users' block of storage will be used for pushdown list as well as data and procedures. For the Q-32, the pushdown list will start at one end of the block and all other storage requirements from the other end; when these meet, it is then necessary either to call the reclaimer or to get more storage from the system. An appropriate modification that utilizes segmentation will be employed on the GE 635.

S-expressions will be represented by list structures as in LISP 1.5. Any kind of entity other than another S-expression can be an atom. Identifiers will resemble LISP 1.5 atoms, and will have much the same kinds of information attached to them (though organized somewhat differently). Character strings, integers, real numbers, and logical values will be organized as blocks where the first word of the block specifies the nature and format of the rest of it. Arrays and matrices will be distinguished; arrays are intended to optimize access to random elements and to permit redefinition of subscripts without moving the actual data elements, while matrices will be used when it is desirable to linearize array element references, as in matric multiplication and computing traces. Syntactically, arrays and matrices will be interchangeable, but one or the other will be more efficient under certain circumstances.

C. Translation Methods

Translation of LISP II source language programs into machine language programs will be based on the use of an intermediate language which will resemble LISP 1.5 and be upward compatible with it. The intermediate language will be capable of expressing all of the features of the source language in more or less one-to-one fashion. It will be highly standardized and much less subject to change than the source language.

Compilation of LISP II source language program will be a two-stage process: a syntax translator will take the source language into the intermediate language, and a conventional compiler will take the intermediate language into machine language. The specification of the compiler will depend on the specification of the intermediate language and the internal conventions of LISP II; the compiler will probably be capable of handling a class of intermediate language programs that contain translations from source language as a proper subset. There may be one compiler built for quick compilation and another built for efficient object code. There will also be an interpreter compatible with the compiler.

The syntax translator will convert LISP II source language programs to the intermediate language at a high rate. It will fill in omitted declarations, supply implicit functions that convert data from one type to another, and post declaration information on property lists of identifiers so that it will be available to the compiler. The programmer may have his own version of the syntax translator with his own special notations added, provided that these notations are properly translated and expressible within the intermediate language. A generator of syntax translators will be supplied with the system. It will be used to maintain the standard translator and to create new ones.

IV MANAGEMENT DATA

A. Corporate Background and Related Experience

Information International, Inc. is an organization of about twenty people specializing in research and development in the computer sciences. The principal office of I.I.I. is located in Cambridge, Massachusetts; an office in New York City is presently being established. Facilities at the Cambridge office include about 8000 square feet of floor space, used for offices, equipment construction, and equipment operation; an electronic workship; a Programmable Film Reader; and an additional computer. The I.I.I. staff includes people with extensive experience in both hardware and software areas; our spectrum of activities ranges from the construction of film reading equipment on a production-line basis to research in artificial intelligence.

I.I.I. was established in 1962 by Edward Fredkin, who is President of the firm. From a one-man operation, I.I.I. has grown to the point where our gross volume of business for 1964 is in the neighborhood of $1 million. I.I.I. is presently organized into four divisions: the Mathematical and Programming Services Division, under Mr. David Lovenvirth; the Behavioral Sciences Division, under Dr. Daniel Forsyth; the Operations Division, under Mr. Goddard Parsons, and the New York Division, under Dr. Paul Abrahams.

The scope of talents available within I.I.I. is best illustrated by the projects that we have undertaken:

(1) LISP

The LISP (LISt Processing) language was devised in 1959 by Professor John McCarthy, then of M.I.T., for use in information processing, symbol manipulation, and programming research. The language incorporates several features such as dynamic storage allocation through list structures, recursion, and the use of conditional expressions that were quite advanced at the time but have since become standard techniques. Since its invention LISP has been constantly improved and expanded, so that it still remains the best and most powerful programming language of its type. For instance, LISP contains both a compiler and an interpreter; the compiler was constructed by expressing it in LISP and interpreting it so that it could compile itself. To our knowledge, LISP is the only programming language in which this can be done.

I.I.I. has been active in the implementation, documentation, and application of LISP. We are presently completing a version of LISP for the Q32 time-shared computer system operated by the System Development Corporation in Santa Monica, California. This system is notable in that the bulk of it was created through the use of a LISP system on the IBM 7090; the compiler was modified to produce Q32 code, and the code thus produced was incorporated with virtually no modification into the Q32 system.

I.I.I. has also been responsible for the publication of the principal book describing the use of LISP. This book, "The Programming Language LISP: Its Operation and Applications", was published by I.I.I. in March, 1964. It includes a simple tutorial section on the use of LISP, articles on the programming system and on debugging techniques, descriptions of LISP implementations on several different computers, and discussions of LISP applications. These applications include the use of LISP as a tool in inductive inference on sequences; a LISP program for checking mathematical proofs; an implementation within LISP of a COMIT-like language for string transformations; LISP as the language for an incremental computer; and the use of LISP in developing an auxiliary input language. Of the fifteen authors of the book, nine have been employed by I.I.I. either full-time as consultants, and much of the work described in the book was performed by I.I.I.

(2) Inductive Inference on Sequences

I.I.I. has performed research on the automation of the use of inductive inference as a means of predicting the behavior of sequences of symbols or numbers. Our work in this area has ranged from theoretical examinations of methods of inference to the implementation of specific techniques on a computer, using LISP. The inductive inference programs are still under development; however, they are already sufficiently sophisticated to predict the next member of a sequence such as

0 0 0 4 0 6 0 8 9 10 0 12 0 14 15 16 0 18 0 20 21 22 ...

in which each prime number has been replaced by zero. They can also predict the next member of more straightforward series such as

Z Y X U V W T S R ...

(The next letter should be O.)

The programs can handle sequences of symbols that can be integers, real numbers, letters, or nonsense syllables. At present, there are two possible output forms: a LISP function that fits the sequence and can be used to predict an arbitrary future member, or a prediction of the next member. The programs are capable of either using or ignoring the conventional meanings of the symbols that they are presented with; those programs that ignore conventional meanings of symbols are first "primed" with collections of sequences that can be thought of as having come from previous experience. The programs will make whatever use they can of these "previous experiences" in their predictions.

(3) Computer Time-Sharing

I.I.I. has been conducting research in computer time-sharing techniques under the sponsorship of the Advanced Research Projects Agency (ARPA). Computer time-sharing refers to a system in which a number of users may simultaneously have access to a digital computer. In operation, the computer alternates rapidly among the programs of the individual users in such a way as to sustain an impression of immediately responsive service for each user. Users gain access to a time-sharing system through individual consoles, consisting of an on-line typewriter or teletypewriter, located in offices and other convenient locations.

I.I.I. has been largely concerned in furnishing research and consulting services to assist in the establishment of a large-scale time-sharing project at System Development Corporation, Santa Monica, California. The time-sharing system there is based on use of an AN/FSQ-32V computer, together with a PDP-1 as peripheral computer for processing input-output. The system will employ about 50 user stations.

In addition, I.I.I. has recently designed a Multi-Processor Computer (MPC-1) specifically for use in a time-sharing system. The MPC-1 consists of several processors and a common memory. The processors are serial, binary machines operating at a clock rate of 20 megacycles. Each contains an accumulator, index registers (one of which is used as the program counter), and boundary registers, all of which are realized in the form of acoustic delay lines.

(4) Programmable Film Readers

I.I.I. has developed a completely automatic computer film reading system (the PFR-1 Programmable Film Reader) which can read film at the rate of approximately 5,000 points per second. Scientific data (e.g., radar data) recorded on 16 mm, 35 mm, or 70 mm photographic film can be read automatically and printed out in the form of numerical listings or recorded on magnetic tape for further processing and analysis. The film reading system is based on three major elements: a general purpose digital computer, together with a visual display scope; a film reading device; and computer programs for using the computer and film reader.

The film reading process itself involves the scanning of film by a rapidly moving light point on the visual display scope. The output of this scanning operation is detected by a photo-sensitive device in the film reader and relayed to the digital computer for further processing and analysis. In addition to translating the data into a more desirable format, additional analysis and summaries of the data may be furnished by the system, as desirable. Film reading systems have been furnished to Lincoln Laboratory and Holloman Air Force Base. I.I.I. also furnishes services for reading films which may be sent to it for processing.

(5) <u>The World Oceanographic Data Display System</u>

A computer-based data display system is under development which will enable an oceanographic researcher to conduct an analysis of oceanographic data using a series of visual displays generated by the computer. The eventual goal of the program will be the design of a computer system capable of storing the entire body of world oceanographic data and making it available for visual display and analysis.

The researcher will be able to select specific oceanographic data for display in the form of contour lines against a map of a selected oceanic area. Data such as water temperature, density, and salinity at specific depths will be available. The researcher will be able to vary the parameters of the data selected, and observe the resulting variations in the contour lines. The system may also be used for the automated production of oceanographic maps and atlases. Specifically, displays in various oceanographic atlas-type formats may be rapidly generated by the computer, and photographs may be taken of these displays which will be suitable for reproduction. It is anticipated that, in the above ways, the world oceanographic data display system will open up the presently available body of world oceanographic data to a more rapid and meaningful analysis by oceanographers.

(6) Computer Display Systems

We are currently completing a project designed to produce a highly efficient, inexpensive computer display system. The display is primarily intended for use in a computer time-sharing system, but is of significant general usefulness as well. The display system is based on the use of a central scan converter servicing a number of individual user displays of the television type. The low cost of the system is derived from the use of television-monitor-type indicators with simple deflection-type circuitry, the use of video storage rather than digital storage, and the absence of complex digital logic in the individual units. In operation, the system has all the advantages of a conventional CRT-light-pen ensemble; i.e., capability for computer-generated displays, light pen interrogation, and writing by means of the light pen on the scope. In addition, the display is flicker-free. The computer itself, however, is utilized for only a small fraction of the time which would normally be required. A low-cost TV camera may be included in the system for visual input into the computer system.

B. Project Management and Personnel

Project MAC will assume responsibility through its delegated representatives for the technical direction of the LISP II project; I.I.I. will assume administrative responsibility. I.I.I. will provide the services of Dr. Paul Abrahams, Mr. Robert Saunders, and Mr. William Henneman for this project. The services of other I.I.I. personnel will be provided as appropriate, with the concurrence of SDC and Project MAC. Resumes of Dr. Abrahams, Mr. Henneman, and Mr. Saunders, together with those of other key I.I.I. personnel, follow.

EDWARD FREDKIN
SUPERVISORY SCIENTIST

Professional Experience: Pilot and intercept controller, USAF 1954-1958; project officer, SAGE Air Defense System for Air Proving Ground and Air Research and Development Commands, (USAF), 1956-1957; Air Defense Project Officer, Lincoln Laboratory (Massachusetts Institute of Technology - USAF) 1957-1958; Lincoln Laboratory (MIT), 1958-1959; Scientist, Bolt Beranek and Newman, Inc. 1959-1961; independent consultant, 1961-1962; President, Information International, 1962-present.

Professional Responsibilities and Projects: Mr. Fredkin has been active in the information sciences, computer design, computer programming, computer languages, and information retrieval since 1956. As an officer in the Air Force (Air Proving Ground Command), Mr. Fredkin was responsible for the planning of data reduction and analysis for tests of the SAGE Air Defense System. Also while in the Air Force (Air Research and Development Command) he participated in a study of the interceptor initiation problem in the SAGE system. At Lincoln Laboratory, Mr. Fredkin worked on problems of pattern recognition and in particular on the problem of similarity. He was also concerned with the problems of efficient utilization of various size computers in organizations conducting research. At Lincoln Laboratory he conducted several computer programming courses and seminars. At Lincoln Laboratory and at Bolt, Beranek and Newman, Inc., he conducted research into advanced information retrieval systems, in particular the concept known as Trie Memory. In addition, he played a leading role in the following projects: the design of a programming system for the LGP-30 computer, the programming of a teaching machine on the LGP-30 computer, the design of a resistor matrix read-only memory, the design of a capacitor matrix read-only memory, the design of the card selector memory, the design of the sequence break system for the PDP-1 computer, the design of the drum system for the PDP-1 computer, the time sharing project for the PDP-1 computer, the design of a large high speed time-shared computer, the FRAP assembly program for the PDP-1, the DECAL Compiler for the PDP-1, and numerous other programs and systems involving various computers. Mr. Fredkin was responsible for the concept and subsequent proposal for a Computer Operated Laboratory, and in that connection designed a video input device for a computer. He was chairman of the DECUS (PDP-1 users group) programming committee 1961-1962; and president of DECUS 1962-1963.

At Information International, Inc., Mr. Fredkin was responsible for the design and programming of the Vortex Ocean Model and the World Oceanographic Display. Under his supervision, a new type of film reader was designed, built, and integrated, along with programs, into a computer system.

Professional Societies: Association for Computing Machinery; Institute of Electrical and Electronic Engineers

EDWARD FREDKIN

Papers and Publications:

"Complex Algebraic Compiler for the LGP-30 Computer", presented at a National POOL Meeting.

"Trie Memory", presented at the Association for Computing Machinery Conference on Symbol Manipulation.

"The LGP-30 Teaching Machine System", presented at the Regional POOL Meeting.

"A Trie Memory Computer", presented at an M.I.T. Computation Center Seminar.

"The DECAL Compiler", presented at the first DECUS Meeting.

"Retrieving Information with an Associative Memory", Association for Computing Machinery Conference on Information Retrieval.

"The Vortex Ocean Model", paper presented at DECUS Meeting.

Invited member of panel on "Machines of the Near Future", paper presented at Gordon Research Conference on "Scientific Information Problems in Research".

"New Directions in Automated Research", paper presented at American Psychological Association National Conference.

"Interaction-Oriented Input-Output Devices", panel on Hardware for Software Types, Fall Joint Computer Conference.

"Computers and Transducers", P.F. Smith and E. Fredkin, ISA Marine Sciences Division, Vol. II, 1962.

"A Time-Sharing Debugging System for a Small Computer", J. McCarthy, S. Boilen, E. Fredkin and J.C.R. Locklider, Proceedings - Spring Joint Computer Conference, 1963.

"Computer-Compiled Oceanographic Atlas: An Experiment in Man-Machine Interaction", Malcolm Pivar, E. Fredkin, and Henry Stommel, Proc. Nat. Acad. Sci., 1963, 50, 396-398.

"Trie Memory", <u>Communications of the Association for Computing Machinery</u>, Vol. 3, No. 9 (September 1960).

"Investigation into the Special Programming Needs for an Automated Laboratory for Psychological Research", Technical Documentary report No. ESD-TDR-63-353, E. Fredkin, 30 May 1963.

"The Time Sharing of Computers", E. Fredkin, Computers and Automation, November 1963, pp. 12-20.

DR. PAUL W. ABRAHAMS
SENIOR RESEARCH SCIENTIST

Professional Experience: S.B. in Mathematics, M.I.T., 1956; Sc. D in Mathematics, M.I.T., 1963; Research Assistant, M.I.T., 1956-1962. Summer and consulting work during a demic training with Remington Rand Univac, Arma Division of American Bosch Arma, IBM, Data Processing, Bolt, Beranek, and Newman, and Bell Telephone Laboratories; Senior Specialist, ITT Data and Information Systems Division, 1962-1963; Supervisor, Information Systems Section, ITT Data and Information Systems Division, 1963-1964; Senior Research Scientist, Information International, 1964-present.

Professional Responsibilities and Projects: Dr. Abrahams' primary field of interest has been in artificial intelligence and in enlarging the capabilities of the computing machine. He has been active in the development of the LISP language and programming system for symbol manipulation and list processing. His doctoral dissertation was entitled "Machine Verification of Mathematical Proof". In this dissertation, he developed a new language, based on LISP, for stating mathematical proofs. Both mathematical logic and systems programming methods were involved in this work. Dr. Abrahams is a member of the American Standards Association X 3.4 Committee on Programming Languages.

Dr. Abrahams has had approximately eight years of experience in the computer field as a Research Assistant at M.I.T., through summer and consulting work, and through his subsequent work at ITT and at Information International. He has programmed for Whirlwind, the IBM 704-709-7094, the IBM 650, the Univac 1103, the Datatron, and the PDP-1. He has specialized in non-numerical computer applications. His computer experience has included: participation in the development of the programming language LISP; design and programming of a compiler for the simulation of the Arma 107A airborne digital computer; participation in the development of a system for program simplification; writing part of a chess-playing program; investigation of languages and techniques for describing computer simulations; design and coding of a generalized computer program for tabulating data from questionnaires; investigation of the General Problem Solver of Newell and Simon; development of a series of data organization programs for the United Nations; and research into quantitative methods of information processing system evaluation. At ITT he supervised the following projects: research into the theory of information retrieval; development of information retrieval systems for patents and medical data; research into adaptive neural networks; and planning of a time-shared multi-user computer system. He is presently responsible for the management of the New York Division of Information International, Inc., and for the development of a computer program for prediction of symbolic sequences.

DR. PAUL W. ABRAHAMS

<u>Professional Societies</u>: Association for Computing Machinery; Institute of Electronic and Electrical Engineers; Professional Group on Electronic Computers of IEEE; Professional Group on Information Theory of IEEE; American Mathematical Society; Institute for Cybercultural Research.

<u>Publications</u>:

"A Flow Chart Synthesis Algorithm", paper presented at the Association for Computing Machinery Annual Meeting, August 1960.

"Machine Verification of Mathematical Proof", Doctoral Dissertation in Mathematics, M.I.T., June 1963.

"Quantitative Methods of Information Processing System Evaluation", Technical Documentary Report No. ESD-TDR-63-670, ITT Data and Information System Division, October 1963.

"Application of LISP to Machine Checking of Mathematical Proofs", in <u>The Programming Language "LISP": Its Operation and Applications</u>, Information International, Inc., Cambridge, March 1964.

EDMUND C. BERKELEY
SENIOR SCIENTIST

Edmund C. Berkeley, Senior Scientist, joined Information International in October 1963. He has been in the computer field since 1939, when he inspected and wrote a report on the Complex Computer made at Bell Telephone Laboratories by Dr. George L. Stibitz. (This machine, using telephone company relays, multiplied or divided complex numbers $(a \pm bj)$ of decimal digits according to an internally stored, constant program.) In 1945-46, Berkeley was at the Harvard Computation Laboratory with Professor Howard Aiken, programming for the first automatic digital computer, the Mark I, and designing some of the circuits for the Mark II. In 1946-48, he participated in the decisions at the Prudential Insurance Company of America, Newark, New Jersey, which led to the first commercial contract for an automatic digital computer. In 1949-50, he and some colleagues designed and built a miniature automatic digital computer using 129 relays, called Simon, which was described in "Scientific American," October 1950. From 1951 to the present, he has been editor and publisher of the magazine, "Computers and Automation."

He is the author of eight books, including: "Giant Brains Or Machines That Think," (Wiley, 1949); "Computers: Their Operation and Applications," (Reinhold, 1956); and "The Computer Revolution," (Doubleday, 1962). He has written over 50 articles, papers, and reports published both in journals and separately.

He is co-editor of the 400-page report, "The Programming Language LISP: Its Operation and Applications," published by Information International, Inc. in 1964; and he is author of the report "LISP -- A Simple Introduction," pages 1 to 49 in that volume.

He is a graduate of Harvard College, 1930, A.B. (summa cum laude; field of concentration, mathematics); a Fellow of the Society of Actuaries (passing 12 professional examinations, 1931-41); and a member of the Association for Symbolic Logic, and the Mathematical Association of America. He was Secretary of the Association for Computing Machinery, 1947-53; he is Editorial Director of the Library of Computer and Information Sciences. He did actuarial work in the life insurance business 1930-48 (except for active duty in the U. S. Naval Reserve, 1943-46; final rank, Lt. Comdr.). He commenced his own business in 1948 which was incorporated in 1954 as Berkeley Enterprises, Inc. This firm is the publisher of "Computers and Automation," and the supplier of the Brainiac electric brain construction kit and several other scientific educational kits designed by Berkeley.

ROBERT A. SAUNDERS
SENIOR PROGRAMMER AND ANALYST

Professional Experience: S.B. in Physics, M.I.T., 1961; Staff Programmer, TX-0 Laboratory, 1961-1962; Senior Programmer and Analyst, Information International, 1962-present.

Professional Responsibilities: While at the TX-0 Laboratory, Mr. Saunders was responsible both for maintenance and modification of the TX-0 and PDP-1 computers and for the development of software for these machines. He was active in the development of the DDT debugging system and the MACRO and MIDAS assemblers, both of which incorporated many advanced techniques. In addition, he wrote a number of library routines for both machines, and served as a consultant on several TX-0 and PDP-1 applications.

Since joining Information International, Inc., Mr. Saunders has been active in several areas. He has developed two film reading systems and has pioneered in the application of signal theory to information retrieval. He has also developed peripheral attachments to the PDP-1 computer for various purposes. At present he is responsible for the implementation of LISP on the Q32 computer at the System Development Corporation; the Q32 LISP system is now operative in a time-sharing mode.

Professional Societies: Association for Computing Machinery

WILLIAM H. HENNEMAN
MATHEMATICIAN AND SENIOR PROGRAMMER

Professional Experience: Completed the course requirements for the Masters Degree in Mathematics at John Hopkins University, working under Dr. W.L. Chow. Currently working on his thesis. Structural designer,Rummel, Kezzer and Kahl, Baltimore, Maryland, 1956-1958; Research Mathematician, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, 1958-1959; Senior Mathematician, Systems Research Group, Mineola, Long Island, 1959-1960, Independent Consultant, 1960-1963; Systems Programmer, Computronics, Inc., Fort Lee, New Jersey, 1962-1963; Mathematician and Senior Programmer, Information International, Inc., 1963-present.

Professional Responsibilities and Projects: As a research mathematician on the staff of the University of Pennsylvania, Mr. Henneman engaged in investigations into various areas of compiler design. These included optimal storage assignment techniques, information retrieval, language translation, etc. This work resulted in implementation of advanced computer systems for LARC and MOBIDIC. While at Pennsylvania, he taught several courses in computing and in applied mathematics.

A past mathematics editor of UNIVAC Engineering Users' Newsletter, Mr. Henneman comes to Information International from Computronics, Inc., where he was a senior systems analyst working on assemblers and compilers.

Mr. Henneman has also consulted independently in various areas, such as operations research, hydrodynamic studies, and statistical decision-making theory.

Prior to this, Mr. Henneman had worked as a civil engineer, stress analyst and bridge designer. During this period, he wrote programs for UNIVAC I involving bridge design and geometry, and the compilation of tables for structural constants of composite metal beams.

Mr. Henneman has developed an input language which translates ALGOL-like language into S-expressions for LISP. His work also includes programming for heuristic retrieval systems and man-machine interactions in the general area of number theory.

Publications:

"An Auxiliary Language for More Natural Expression - the A-Language", in The Programming Language "LISP": Its Operation and Applications, Information International, Inc., Cambridge, Massachusetts, 1964.

V. PRICE AND PAYMENTS AGREEMENT

Information International, Inc. agrees to perform work towards the development of the Lisp II Programming System described herein for an initial period of 10 months at an estimated cost plus fixed fee of $133,500. It is assumed that progress payments will be made on a monthly basis, based on costs incurred. This price is broken down on the following page.

BREAKDOWN OF ESTIMATED COST

Direct Labor		
Senior scientists	$10,000	
Supervisory programmers	6,600	
Senior programmers	27,500	
Programmers	3,400	
Total direct labor		$ 47,500
Overhead @ 125%		59,375
TWX Charges		12,000
Outside Purchase		
Miscellaneous Computer Time		1,000
Travel		
6 round trips, Boston-LA-Boston @$304	$ 1,824	
10 round trips, Boston-NY-Boston @$27	270	
Per diem, 100 days @$18.00	1,800	
Total travel		3,894
Total Cost		$123,769
Fixed Fee		9,731
TOTAL COST PLUS FIXED FEE		$133,500