Research Proposal Submitted to the National Science Foundation

Proposed Amount 113,544 Proposed Effective Date 6/1/76 Proposed Duration 24 (months)

Title MOLGEN: A Computer Science Application to Molecular Genetics

Principal Investigator

Soc. Sec. No.

Institution
(if different from submitting institution)

Address

Branch/Campus

Submitting Institution Stanford University

Department Computer Science

Branch/Campus Main

Address Stanford, California 94305

Co-Principal Investigator Edward A. Feigenbaum Soc. Sec. No. [redacted]

Co-Principal Investigator Joshua Lederberg Soc. Sec. No. [redacted]

If renewal request previous NSF grant No.

Make grant to Stanford University
(name of institution or organization to which grant should be made)

Endorsements:

	Principal Investigator(s)	Dept. Head	Institutional Admin. Official
Name	Edward A. Feigenbaum	Robert Floyd	D'Ann Downey
Signature	Edward A. Feigenbaum	Robert W Floyd	
Title	Professor of Computer Science	Chairman, Department of Computer Science	Asst. Sponsored Projects Officer
Telephone Number	(415)497-4878	(415)497-2274	(415)497-2883
Date			

Research Proposal Submitted to the National Science Foundation

Proposed Amount 113,544 Proposed Effective Date 6/1/76 Proposed Duration 24 (months)

Title MOLGEN: A Computer Science Application to Molecular Genetics

Principal Investigator

Soc. Sec. No.

Institution
(if different from submitting institution)

Address

Branch/Campus

Submitting Institution Stanford University

Department Computer Science

Branch/Campus Main

Address Stanford, California 94305

Co-Principal Investigator Edward A. Feigenbaum Soc. Sec. No. [redacted]

Co-Principal Investigator Joshua Lederberg Soc. Sec. No. [redacted]

If renewal request previous NSF grant No.

Make grant to Stanford University
(name of institution or organization to which grant should be made)

Endorsements:

	Principal Investigator(s)	Dept. Head	Institutional Admin. Officia
Name	Joshua Lederberg	Joshua Lederberg	D'Ann Downey
Signature			
Title	Professor & Chairman Department of Genetics	Professor & Chairman Department of Genetics	Asst. Sponsored Projects Officer
Telephone Number	(415)497-5801	(415)497-5801	(415)497-2883
Date			

1 PROLOGUE

The scientific text of this proposal is identical with that of the companion from the University of New Mexico. The budget is separate and it will be administered by Stanford University as indicated in the budget page. We realize that this is an unusual procedure but we feel it is justified as a means of bringing geographically separate scientists together to work on common problems. The personnel involved in the proposals began working together on the organization of the project in June, 1975 when Prof. Martin visited Stanford for the summer. We are proposing that Martin spend the major part of the first year at Stanford, working closely with the Stanford group on all phases of the foundation of the system. This is strongly supported by Prof. D. Morrison, chairman of the Division of Computing and Information Science at UNM. The second year Martin will return to UNM to continue working on special subparts of the system with the aid of a UNM graduate research assistant. Constant communication can be maintained through the excellent SUMEX-AIM time sharing installation as outlined in the Resource Section of this proposal. At regular intervals all participants in the project will meet at Stanford to review progress and suggest new directions.

Table of Contents

Section   Page

Subsection

1 Introduction

The general goal of the research proposed here is to develop a computer program (termed MOLGEN) to assist a molecular geneticist in planning laboratory experiments. It is to be an interactive system, drawing on both the expertise of the human geneticist and the expert knowledge stored in the data base. The system must be convenient and comfortable for the geneticist, and will need to be sophisticated and powerful from the point of view of computer science. The program will itself be an experiment in the development of intelligent systems.

The need for such a program is suggested by the recent growth of technology in molecular genetics. In the past three years many new site-specific restriction enzymes have been discovered (Nathans 1975). These enzymes have been applied to physical mapping of chromosomes, nucleotide sequence analysis of DNA, isolation of genes, and restructuring of DNA molecules. Such rapid growth extends throughout the field, with continuing advances in separative and physical techniques. With the many new procedures, there are possibilities for novel combinations of techniques which broaden the horizon of possible experiments. At the same time, it has become increasingly difficult for any one scientist to keep track of what is available, as well as the limitations and use of the newer techniques. Many "ingenious" discoveries, in fact, can be viewed as a judiciously selected combination of well established unit processes. Hence an intelligent computer system is likely to be pragmatically useful in this area.

An immediate application of the proposed systen will be the simulation of recombination of random segments of bacterial DNA cut by restriction endonucleases. In experiments currently underway in Professor Lederberg's laboratory, geneticists would like to have quantitative predictions of reaction component concentrations at any particular time, given initial concentration of Bacillus subtilis DNA and site-specific restriction enzymes. The conception of the enzyme simulation program (see part III) in the MOLGEN system was directed precisely toward this kind of need. Part of this effort will involve the creation of a knowledge base of restriction enzymes with associated kinetic data, recognition site information, and characterization of availability and degree of purity. Convenient access to a comprehensive source of this information would itself be of considerable value.

A subproject which should also prove very useful to the geneticist will be the summarization of all empirically established sequences of DNA and RNA, along with restriction enzyme site specificity. Such information might play an important role in the current effort of Professor Lederberg's laboratory in dissecting the genetic structure of Bacillus subtilis. Recognition sites in a wide variety of naturally occuring nucleic acid species has been summarized in a preliminary way by Sobell (Sobell 1973).

An important factor in selecting this domain was the availability of a great many tangible problems covering a wide range of difficulty. This means that it will be possible to develop the system incrementally, and still be able to perform useful tasks even at the outset.

That branch of computer science known as artificial intelligence (AI) has already developed techniques for computer-assisted problem solving in the experimental sciences. Examples of successful projects are: the DENDRAL project at Stanford in the field of mass spectrometry; the organic synthesis systems developed by Wipke, Corey, and others; the MYCIN project at Stanford in the field of chemotherapeutic advice; and Pople and Myer's DIALOG system for medical diagnosis (Pople 1975). The joint effort of computer scientists and experts in the experimental sciences has allowed the creation of excellent computer systems. These systems are presently being used to facilitate research in the experimental sciences. The ideas and technology behind the projects are now available for application to an experimental planning program for molecular genetics.

There appear to be two factors common to those task domains to which computer-assisted problem solving techniques have been applied successfully:

1) the development of systematic and logical methods of analysis of problems and application of tools by the experimental scientists themselves,

2) the possibility of solving problems in the domain using a strongly constrained and consistent body of knowledge.

In the current state of the art, heuristic problem solving programs containing a few thousand facts and heuristics about a task domain can be built and used effectively. The existence of these two factors in a domain suggests that the computer scientist can build an intellectual bridge into the domain in a period of time that might be measured in months rather than years.

Some problems of experimental molecular genetics currently share both of these factors. For example, the growth of knowledge about restriction enzymes has been important to genetics because of their many applications to manipulation and separation of DNA structures. Despite the large number of enzymes, each can be represented by a small set of descriptors. This makes the task of representing the knowledge far more limited than it might first appear.

One major issue of artificial intelligence that relates to this research is how to plan experiments involving the choice of several steps in a large space of possibilities. This includes such fundamental points as when to design an entire experimental plan first and then fill in details later (top-down), and when to design a complete step at a time (bottom-up). A second major issue will be the development and management of large knowledge bases. The MOLGEN system will need extensive files of knowledge about DNA structures, enzymes, separative techniques, and physical processes. For example some of this knowledge will involve mathematical models of kinetics; some will be detailed sequences of base pairs; some will be "fuzzy" and inexact descriptions of a particular process or structure. The knowledge bases must be designed to allow as much of the data as possible to be handled in a uniform manner. A third major computer science problem will be the creation of a simulation program which can simulate the action of enzymes on DNA structures. Many of the current techniques developed for simulation in other domains may prove useful here. The MOLGEN system will also have to deal with cases of incomplete and inexact knowledge in the planning and the simulation stages. Molecular genetics, as a task domain, offers problems which are limited and structured enough to be feasible with available techniques, as well as those which are challenging enough to advance the art.

The overall purpose, then, of providing computer assistance in the planning process is to make it possible for an investigator to explore a wide range of possible experiments. Calculations of the effectiveness of computer assisted planning can only be approximate, but the following is offered for a rough estimate of the potential. Professor Lederberg has suggested that the knowledge actively used in the planning of experiments in molecular genetics could be encoded as approximately a thousand computer "rules". (This fits confortably in the range used by current knowledge-based intelligent systems.) It has also been observed that most experiments in molecular genetics involve on the order of seven levels of formal decision once the context of a particular goal has been set. We believe that fairly crude heuristics can reduce the likely choices to perhaps ten. Ten major alternatives at each level of planning yields a total "search space" of about ten million alternatives. A search space of this size fits within the limits of current AI programs but still demands a definition of more powerful search-reduction heuristics for high performance. The purpose of these estimates is not so much to give a true measure of program performance, as to illustrate the notion that the number of possible experiments is indeed much greater than could actually be performed. Prof. Lederberg has suggested that a number of important and perhaps novel experiments are not being considered because of the difficulties in keeping current in a rapidly expanding discipline. Meeting this need is precisely the goal of the MOLGEN project.

2 Background and Related Work

2.1 Relationship to Other Research in Artificial Intelligence

There are two primary reasons for proposing the MOLGEN project:

(1) to develop a high performance computer program augmenting human intelligence in important research areas in molecular genetics,

(2) to provide a context for testing and extending ideas in the design of intelligent systems.

This section will discuss the existing scientific and technological base and some proposed extensions and show the relationships between MOLGEN and other AI programs.

2.1.1 Planning and Heuristic Search

The problem of planning an experiment fits into the paradigm of heuristic search, a well established framework for computer based problem solving. Problems in this framework can be represented as a tree of subproblems, with solutions existing at unknown depths along unknown paths. Judgmental rules and procedures, called heuristics, are applied to direct the search. When we speak of the expertise or good judgment of an expert, we are often referring to the heuristics he has developed to search effectively.

There are typically a number of alternative procedures to choose from at each step in the course of an experiment in molecular genetics. These alternate steps can be viewed as a family of possible experiments. This family may then be represented as a tree structure so that the planning of a particular experiment is equivalent to choosing a path in the tree. Each node represents a certain "state" in the experiment where the physical entities being used by the experimenter are in a particular condition and certain quantities are known. For example, the physical entities may be a mixture of oligonucleotides whose terminal base sequences are known. The activity in the planning process centers around selecting an operation to carry the experiment to the next state. These operations are chosen from the program's knowledge base and are naturally expressed as transformations from one state to another. For example, in the the context of molecular genetics, a chromatography step could be represented as a transformation which "transforms" a mixture (initial state) into its separated component parts (final state). Each particular type of chromatography would be more specifically described by the parameters of the physical entities which are active in the separation. For example, gel electrophoresis would be described in terms of its ability to separate oligonucleotides according to length. Part of the knowledge embedded in the transformation would be a rule for estimating the effectiveness of gel electrophoresis on any particular mixture of oligonucleotides for the various gels which are available. Finally, a completed plan is a sequence of transformations which when applied successively to the initial state of the experiment transform it to the desired goal or final state.

This model of problem solving based on the selection of a path according to heuristic knowledge is often called the heuristic search paradigm. It has been used as a model for a variety of problems since such early projects in AI as the "Logic Theory Machine" (1956) by Newell and Simon. One dimension of progress in AI has been the development of techniques for the creation of plans within the heuristic search paradigm.

2.1.2 Strategies for Planning

Several algorithms have been developed to assist in choosing a "minimum cost" path in a heuristic search tree (Nilsson 1971), where an estimating function is available which can measure how "close" any intermediate point in the experiment is to a final state. In order to guarantee that the algorithms will find a minimal path, the estimating function must never overestimate the distance to the goal. For complex problems, a practical difficulty continues to appear in many contexts. Simply stated, it is sometimes best to retreat from a goal in order to get closer to it. In mathematical theorems, this arises in those cases where it is easier to solve a more general theorem than a specific one. In organic synthesis, it is sometimes better to build up a rather complicated structure which seems "farther" from a target compound than some current step in the synthesis, but from which a direct and often elegant reaction will transform the complicated structure almost directly to the desired product.

In complex scientific domains, the distance estimating functions are almost impossible to find. Even if such a metric can be found, it is unlikely we can guarantee that it never "overestimates" the distance to a goal state. Nevertheless, in special cases, quantitative distance measures can sometimes be found when discrimination among the products of the alternative transformations is statistically testable. Otherwise, planning programs must rely on domain specific or even goal specific knowledge to guide the selection of the next step.

Some more general techniques for planning have been developed for robot problem solvers. For example, the STRIPS system (Fikes 1972), in the course of building long plans, attempts to assemble short sequences of operators which may later prove useful. The idea is that it is more efficient to assess the effect of the smaller plans and reuse them than to generate them again from scratch later. The algorithms include methods for disentangling the local effects of that piece of a plan from its immediate context and predicting its effect in a new context.

Another planning idea of general application has been developed in Sacerdoti's work on hierarchical planning (Sacerdoti 1973). This can be seen as a formalization of the common sense notion of planning major steps first, and attending to details later. In our terms, it means recognizing that some preconditions for use of a transformation are more important than others. This ordering allows the most important criteria to be considered first. In contrast to earlier methods for generating plans, which proceeded a step at a time toward a goal, hierarchical planning involves sketching out a plan in several passes of refinement of detail. This approach is applicable where it is possible to indicate an ordering for the criteria used in choosing the next step of a plan.

Our examination of several experiments in molecular genetics suggests that this idea has a great deal of relevance. For example, in dealing with enzyme data, it is far more important for most purposes that an enzyme be applicable, i.e. that it have the desired effect on a substrate, than that its most active pH be in a certain range. In terms of planning, this suggests that "applicability" is a more important piece of planning information than is "pH range". Hence, the adjustment of pH in an experiment is a detail which can be considered during a later pass in the planning process. This ordering of importance need not be assumed constant for all experiments, but may be considered as part of the planning heuristics. The application and development of these planning issues will be part of the interesting computer science questions to be investigated in the MOLGEN project.

The problem of experiment planning can be compared to the automatic generation of computer programs. The planner's task is to find a sequence of instructions that will transform the input into the desired output. However, the tools that the experiment planning program will work with are conceptually simpler and the range of possible "programs" or experiments is more restricted than those utilized in the general problem. Although there are many unit processes, the planning will be facilitated by needing little or no recursion or looping, limited "data types", few steps, and in addition will have a detailed knowledge of the subject matter. Thus the heuristics necessary to plan experiments are more manageable.

2.1.3 Acquisition of Knowledge

One trend in the development of intelligent systems has been an increase in the size of the knowledge base used by the programs. Domain specific knowledge is used to guide and hence speed their search. The process by which a system acquires this knowledge is one of the central design issues for an intelligent system. The DENDRAL programs, for instance, which are used in the analysis of mass spectra of organic compounds (Buchanan 1969), acquired their chemical and mass spectral knowledge by a painstaking process in which an AI scientist worked with a chemist, eliciting from the chemist the theories, facts, rules, and heuristics applicable to reasoning from mass spectra. This same approach has been used successfully in protein structure determination using x-ray crystallography and in diagnosis of glaucoma eye disease (Kulikowski 1973).

More recent versions of the DENDRAL program have special built-in editor programs for acquiring knowledge and data from the chemical domain. For example, a chemical structure editor is used in entering fragments of chemical structures, and other editors specialize in acquiring chemical structure constraints and fragmentation rules. Another approach, typified by the SECS program for organic chemical synthesis (Wipke 1975), involved use of a special language (ALCHEM) for describing organic reactions. For each of the areas of organic synthesis expertise, a chemist specialist was responsible for encoding in ALCHEM the particulars about the useful reactions in his specialty. The idea was that a user of the SECS program could then draw upon the combined knowledge from all the experts who had participated. This process not only contributed to the growth of the knowledge base, but also resulted in changes and additions to the ALCHEM language which made it both more powerful and more convenient.

The MYCIN program, used for diagnosis and treatment planning of infectious disease using antibiotics (Shortliffe 1973, 1975) uses a more elaborate method for the acquisition of knowledge. In this system, knowledge is embedded in a collection of some 300 decision rules. New rules can be added via a rule acquisition program. This program accepts rules in a stylized form of English, converts then to an internal form, and then prints them out again in English to indicate its understanding of the rule. The system also has a model of its knowledge, in the form of abstracted descriptions of classes of rules. It uses these both to help interpret the rule the physician has typed in, and as a check on its content. If, for example, the new rule differs from most others in that class, the physician is warned, and can then alter the rule if he wishes.

One of the more interesting components of the MYCIN program is its explanation system. This concept was introduced earlier in

Winograd's SHRDLU program (Winograd 1972), and refers to the ability of a program to explain its activities to its user. For example, MYCIN may ask the physician a particular question about the results of a culture. If the physician wants to know why the question was asked, he can ask the system. It will then examine its reasoning, and indicate the purpose of the question by displaying the relevant rule. The same rules which guide the consultation are thus being used to explain the program's behavior.

2.1.4 Applications to MOLGEN

It is our goal in the development of the MOLGEN system to draw upon and extend some of the ideas and techniques mentioned above. With regard to planning, we will be experimenting with ways to relate hierarchical notions of planning to the concepts of molecular genetics. Furthermore, we will investigate techniques to make the MOLGEN programs handle the diversity of goals in molecular genetics experiments. Note that when a physician starts to use MYCIN, the program does not have to ask him what he wants to do. The physician always wants to diagnose a disease and prescribe treatment. For MOLGEN, however, the experimental goals are as diverse as the experiments in molecular genetics.

We expect MOLGEN to draw upon the same knowledge base in its efforts to work on a variety of experimental planning problems . In spite of the diversity of experiments in molecular genetics, we have categorized three main types of experiments as follows:

1. Given a structural hypothesis, plan experiments and

procedures which would verify the hypothesis. (This is an example of structural analysis.)

2. Given starting and target DNA structures, plan experiments and procedures which will transform the initial structure to the final structure. (This is an example of structural synthesis.)
3. Given an initial DNA structure, hypothesized final structure, and a set of steps in an experiment to be carried out, verify the probable correctness of the final structure or show any other possible final structures. (This is an example of a 'proof' checker.)

An initial goal towards meeting the first two objectives is the development of a planning program which will allow a user to "sketch" out an experiment, and allow MOLGEN to fill in the details . An example showing how this might proceed is given in section 3 of this proposal.

A second area for developing on the themes mentioned above is in the area of knowledge acquisition and representation. The main extension to previous ideas has to do with the handling of a diversity of data types. MOLGEN must deal with a broad range of types of knowledge -- including DNA structures, enzymes, chromatographic techniques and various physical processes. For each of these types of data, MOLGEN is expected to assist the user in entering and modifying the data and knowledge base. This should be done in a form convenient to the user's view of each particular type and yet at the same time in a way which keeps the program's diverse collection of knowledge and data types from becoming too complex to be manageable. Similarly, the explanation system must be implemented in a way that makes the reasoning of the program accessible to the user, through all levels of the knowledge base. A successful treatment for a broad spectrum of diverse knowledge will be a major design goal for the MOLGEN project.

2.2 Related Work in Computers Applied to Genetic Biochemistry

We propose to provide in this section a summary limited to work directly related to the simulation and planning aspects of the MOLGEN project.

Some of the most advanced prior work in simulation has been done in the area of enzyme kinetics. The basic approach has been to set up the differential equations describing the chemical reactions , and then solve those equations on either a digital (Chance and Shephard 1969, Bates 1973, and Garfinkel 1968) or analog (Chance 1967) computer system. It is interesting that attempts have been made to facilitate use of the systems by working geneticists, either by means of interactive display mechanisms (Bates 1973), or an English-like language for describing the chemical systems (Garfinkel 1968). We intend to expand upon these user-oriented techniques during our system development.

We also know of a system (Green 1970) which uses matrix representations of the steady-state equations describing enzymes to permit rapid calculation of the distribution of all enzymatic forms at any given moment. We feel that the accurate modelling of this steady-state behavior will prove vital to our enzymatic simulation system, although we still have to investigate the general applicability of the matrix representation technique for multi-enzyme systems.

Crothers (1968, 1971) has developed several computer programs to simulate various aspects of the process of de/renaturation. The detailed mathematical models available on conformational changes (Bloomfield 1974) will facilitate the computer simulation of these processes.

The problem of representing the often "fuzzy" knowledge about DNA structures has been considered in a model for the heterogeneity of base composition (Elton 1974). The point is that the exact composition is rarely known for more than a few hundred bases. Elton's model classifies the DNA structures into "segments" with different underlying base compositions, each segment categorized by an estimate of length and G-C content. Our representation model for DNA structures will allow such classification of portions of DNA, as well as many other factors like the distribution of important features (nicks, gaps, etc.) along parts of the DNA strands.

Finally, we find of direct relevance to our design of experimental plans the use by Khorana's group (Powers 1975) of computer programs to help plan the detailed steps involved in gene syntheses.

The genes are synthesized in small segments and then joined by some chemical or enzymatic means. The program, using knowledge of the chemical environment, determines optimal segment lengths and order of joining. This will be one of many experimental paradigms we hope the MOLGEN system will encompass.

2.3 Supporting Research at the Stanford Heuristic Programming Project

The Heuristic Programming Project at Stanford, of which the MOLGEN research will be a part, is one of the few AI research groups that emphasizes development of intelligent systems for use in the sciences. This section discusses some of these intelligent systems which are being developed and actively used. These interdisciplinary projects serve to illustrate the range of experience that the MOLGEN research can draw on.

2.3.1 The DENDRAL Project

The Heuristic DENDRAL project was started by Professors Lederberg and Feigenbaum in 1965, and has diversified and matured considerably since that time. The DENDRAL program (Buchanan 1975 and Appendix A) is an application of artificial intelligence to biomedical molecular structure determination problems. The program is designed to explain empirical data, in particular, to interpret experimental data from a mass spectrometer, an analytical instrument used in organic chemistry. It uses heuristic search techniques to explore plausible candidate explanations of the empirical data in much the same way that other artificial intelligence programs explore spaces of hypotheses.

A major contribution has been demonstrating that a large amount of scientific knowledge can be managed and used by a complex problem-solving program. Another contribution has been showing that AI techniques can be successfully applied to scientific problems.

The DENDRAL molecular structure building program, CONGEN, is currently used by chemists as a tool for building molecular structures from structural pieces which the chemist has inferred from whatever data are available. The DENDRAL PLANNER (Smith 1973) uses a large amount of knowledge specific to mass spectrometry to infer structural constraints directly from empirical data. The DENDRAL PREDICTOR is a simulation model of the mass spectrometer that allows the program to make testable predictions of candidate hypotheses. META-DENDRAL is a program for discovering new inference rules in the field of mass spectrometry.

From the start, major emphasis has been placed on avoiding the "outsider-insider" distinction by having trained chemists work on the programs and make them useful to working scientists. This emphasis is in contrast to other research on "toy" problems conducted without reference to the needs of working scientists. Since May 1974, the project has moved to the interactive computing environment of the NIH-funded SUMEX-AIM facility (Carhart 1975). Because of this, many scientists outside this university have been able to use the DENDRAL computer programs to further their own research. These programs are currently receiving heavy use from local users and outside users who are investigating mass spectrometry problems for a variety of different compound classes.

2.3.2 The MYCIN Project

The MYCIN project (Davis 1975) at Stanford is another interdisciplinary research group including members of the Heuristic Programming Project, whose goal is to apply artificial intelligence reasoning programs to a scientific problem. The objective of this MYCIN research is to develop a computer-based system capable of using the judgmental knowledge of experts to assist physicians in selecting appropriate antimicrobial therapy for patients with infectious diseases. The work concentrates initially on the use of antimicrobial agents in the treatment of bacteremias.

The current MYCIN program utilizes data available from the microbiology and clinical chemistry laboratories, plus the physician's response to computer-generated questions, to provide physicians with consultative advice on antimicrobial therapy. Therapy selection takes into account both the patient's infections and the range of possible identities of the organisms causing these infections. One contribution of the MYCIN system has been developing general techniques for explaining its reasoning steps to permit the program to explain all of its actions and reasoning, including, for example, the deduction of the identities of pathologic agents. Another important part of the MYCIN system is a rule-acquisition program for computer acquisition of judgmental knowledge about those concepts which the program uses in making deductions. This permits experts in the field of infectious disease therapy to teach the system those therapeutic decision rules which they use in their clinical practice. Making rules, exploring MYCIN's reasoning, or adding new knowledge to the system is done in a stylized subset of English.

2.3.3 Protein Structure Modeling Project

The Protein crystallography project involves scientists at the University of California at San Diego as well as members of the Heuristic Programming Project. The general objective of the project is to apply problem solving techniques in artificial intelligence to the "phase problem" of x-ray crystallography, in order to determine the three-dimensional structures of proteins. Specifically, the task is to develop a computational system which can infer the tertiary structure of a protein molecule in the absence of phase information normally obtained from multiple isomorphous replacement procedures.

In the context of artificial intelligence, project objectives center around knowledge acquisition and program organization. The knowledge acquisition task involves formalizing the knowledge and heuristics used by expert protein crystallographers to infer the tertiary structure of proteins from x-ray crystallographic data, and to formalize this expertise in appropriate data structures and heuristic procedures. The program organization plans are variations on the

HEARSAY (Reddy 1973) model of cooperating "expert" modules in a program.

3 Detailed Plans for Work During the Proposal Period

3.1 Introduction

This section presents our overall view for the molecular genetics system as well as a detailed list of specific goals for the first year covered by this proposal. The system will be composed of three major, interacting parts: an experiment planning system, an enzymatic action simulation program, and a collection of knowledge bases containing the rules and heuristics of molecular genetics.

The experiment planning program will collect information about a problem from the user, select an appropriate methodology (information retrieval, simulation, hierarchical planning, or some combination) and then work interactively with him to solve the problem. Some questions can be answered by information retrieval from a knowledge base, such as a question about the enzymes that function under certain pH or salt concentrations. On the other hand, if a geneticist asked for a prediction of the ratio of linear to circular DNA in a test tube after 10 minutes application of ligase, then a straightforward simulation would be in order. In verifying that a given set of steps in an experiment produces the final result, a combination of simulation and retrieval would be used by the planning program. In more difficult, higher order problems, such as structural analysis or synthesis, hierarchical planning is needed, perhaps involving both retrieval and simulation as well.

A simulation program will provide detailed modelling of the action of enzymes on DNA structures. Our goal is a program which will be able to produce accurate enough results to answer questions like the one posed above, i.e. what will happen after application of ligase for 10 minutes, 2 hours, etc.

The knowledge bases will be composed of collections of the rules and heuristics used by geneticists, as well as facts about enzymes, experimental methods, and physical processes like renaturation. They must allow access in retrieval, simulation, and planning modes, so provision for the representation of many types of knowledge must be supplied.

Along with the major system components discussed above, certain themes will remain dominant during all phases of system design. Primary consideration will be given to making the system an easy and natural tool for the geneticist to use. DNA structure entry, editing, and display will be by way of an interactive program with numerous user-oriented features. Explanation facilities (in the model of the MYCIN system) will be provided whenever possible, and all knowledge bases will be made easily extendable and modifiable. Building the trust of the user is considered necessary to the continued development of the system. The best way to provide for this will be to insure that it is used by geneticists, and that a strong interaction between computer scientists and genetics experts continues.

The remainder of this part of our proposal will discuss in detail our plan for system development in the first year. Briefly summarized. our four major goals are:

1. Begin the construction of a rudimentary experiment planning program and an associated planning knowledge base.
2. Build a simulation program which will apply basic enzymatic actions to DNA structures and summarize the resulting structures.
3. Design the knowledge base containing rules and heuristics used by the geneticists, enzyme knowledge, and knowledge of experimental methods and physical processes.
4. Complete the construction of an interactive DNA structure entry and editting system.

3.2 An Experimental Planning Program

The central, and most interesting part of the Molecular Genetics project will be the experiment planning program (PLANEX). PLANEX is meant to be an interactive program which combines the intuition and expert knowledge of a molecular genetics investigator with the thoroughness of a computer having a detailed knowledge base. The investigator will sketch the initial conditions for an experiment and the desired final condition. PLANEX will be developed to allow the user to specify required or suggested intermediate steps. PLANEX will suggest intermediate steps, additional options, and verify the expected results within the limits of the knowledge base The program will initially be designed to check the steps of an experiment, and possibly fill in the details between small steps. The direction of development of PLANEX will be toward a program which can eventually take bigger steps, interpret less precise requirements by the experimenter, and offer more useful alternatives based on the knowledge base. The investigator could request varying degrees of detail and, at all times, the heuristics and reasoning tools used by PLANEX as it evaluates alternatives would be accessible to a user via an explanation system. Freed from the constraints of checking all details, the experimenter could explore the possibilities of many experiments before choosing one and also have novel experiments presented for his consideration.

A scenario for a possible run of PLANEX might be the following. We suppose that a molecular biologist has a new restriction enzyme (call it R12) and that he wants to consider alternative experiments for determining its specificity, i. e.. the specific site on the DNA molecule for its application. His first step might be to create an enzyme description of R12, using the procedures for entering information about any enzyme in the MOLGEN knowledge base. The description would include such information as its name, enzyme classification ("endonuclease" if nothing more specific is known), IUPAC number, cost, availability, stability, salt activity tables, substrate description, names and concentrations of impurities known to be present.

When the available information about the new enzyme was entered, the user would then call in PLANEX. He would tell PLANEX that he wanted an experiment to determine the specificity of the enzyme R12. PLANEX would ask if the user has digested some DNA to exhaustion and determined the initial and final segment sizes. (From this, the length of the restriction sequence can be estimated. We assume that this length is estimated to be five nucleotides.) PLANEX now interprets the user's experiment as the following:

Given the initial state as follows:

Initial State: Segments of unknown base sequence (resulting from complete digestion of phage DNA by R12).

construct a sequence of steps to the final state where:

Final State:

1. The identities of the last few nucleotides on the 5′ ends of the fragments have been determined.

2. The identities of the last few nucleotides on the 3′ ends of the fragments have been determined.

For simplicity, let us assume that the user is willing to limit his initial goal to determining the identity of the terminal nucleotide and that he wishes first to do this for the 5′ ends of the fragments. The choices for doing this include

A) Label the 5′ termini of the fragments with radioactive phosphate groups followed by a separation procedure,

B) Convert the 5′ end to hydroxyl using phosphotase. The terminal base can then be distinguished from the other nucleotides by chromatographic means after a 3′ to 5′ exonuclease digestion.

Successful reasoning by PLANEX at this level will depend on characterizing the available options. The more general the classifications and heuristics, the more apt PLANEX will be at generating new combinations of techniques. For example, the two methods of terminal nucleotide analyses mentioned above would fit into the following general scheme:

To determine the identity of the 5′ terminal nucleotide on an oligonucleotide,

1) "Label" the end nucleotide.

2) Break the oligonucleotide into pieces which can be separated.

3) Identify the pieces which are labeled.

In this context, a "labeling" means any technique which makes the piece containing the terminal nucleotide distinguishable in some separation and identification procedure. It would include the above techniques as well as, for example, replacement of the terminal base in a predictable way by a base analog (as in the "turnover" technique using Polymerase).

Let us suppose that one of the experiments that the user wants to consider at this point is method (B) from above, that the Snake Venom 3′ to 5′ exonuclease has been chosen to break the oligonucleotide into pieces, and that the separation technique is a type of chromatography capable of distinguishing nucleotides from nucleosides and determining their identity. The experimental plan at this point looks like the following:

Initial State: Mixture of oligonucleotides of average length 200 nucleotides with unknown 5′ terminal nucleotides.

Operation: Apply Phosphotase.

State: Mixture of oligonucleotides of average length 200 nucleotides with unknown 5′ terminal nucleosides.

Operation: Apply Snake Venom 3′ to 5′ exonuclease.

State: Mixture of nucleotides and (terminal) nucleosides.

Operation: Separate nucleosides from nucleotides and determine identity of nucleosides.

Final State: Nucleosides have been identified. (Identity of 5′ terminal nucleotides of fragments has been determined.)

At this point, the experiment is thoroughly outlined, although there are a number of smaller steps still to be determined. The user asks PLANEX to fill in some more details. This means that PLANEX should generate the intermediate steps so that the "required input" for each operation is matched by the "output" of the previous step, that is, so that there is a complete sequence of states and operations from the initial state to the final state. In this case, PLANEX suggests the use of Pancreatic endonuclease after the Phosphotase step and before the Snake Venom step to reduce the length of the oligonucleotides as required for more rapid action by the Snake Venom exonuclease. Similarly, a denaturation step may be inserted before the Phosphotase step. The generation of both of these steps is caused by the interpretation of the enzymatic knowledge base for the enzymes used in the operation. At a finer level of detail, PLANEX will consider steps which adjust the pH or ionic concentrations to maximize the reaction yield. Heuristics, under user control, weigh the various considerations which lead to the generation of these subgoals into a hierarchy - so that the "more important" criteria are considered first. Finally, the user may ask PLANEX to estimate the yields, costs, and time required to perform the overall experiment.

At any point in a session, a user could backtrack to explore a different possibility. The ability to compare several different experiments is useful in cases where confirming experiments are used to guard against experimental error. In many cases, planning would not proceed to the end of an experiment -- as when the results at a particular step dramatically affect the selection of the following step. An example of this occurs in the prologue of the scenario experiment, when PLANEX asked the user for information necessary to estimate the length of the recognition sequence of the enzyme. Had the user elected to determine more of the sequence than the end nucleotide, this information would have been essential in choosing between methods which use overlapping sequences. The user's choice to identify only the end nucleotide greatly simplified the experiment.

3.3 An Enzyme Simulation Program

Enzymes form the primary tools geneticists use to manipulate DNA structures. The most common types include exonucleases, which break the backbone phosphodiester bond starting from an end, gap, or nick; endonucleases, which break a internal backbone bond; ligases, which seal a break in the DNA backbone; and polymerases, which add bases to a primed single DNA strand and fill in gaps in double strands. As mentioned, a special type of endonuclease, the restriction enzyme, functions to break the DNA backbone at very precisely specified sites. All of these processes must be simulated to provide accurate modelling of enzymatic action. One of the first processes that we will model will be the ligation of endonuclease-generated DNA fragments into linear and circular structures.

The simulation program will operate in the following manner. The program is given the detailed action to be carried out (e.g. apply a 3′ to 5′ exonuclease) and the initial pool of the various types and concentrations of DNA structures present. It will decide, using advice from the user, what structural features are important in this experiment, and focus on those types as the simulation proceeds. The program will choose an operator function and apply it to a structure selected stochastically from the pool, producing a possibly new structure. This may either increase the concentration of one of the present structures (decreasing that of another) or add a new structure to the pool. The process will continue until all structures are inert to enzymatic action, or until specified time interval has passed.

One major representation difficulty for the simulation program is that the number of DNA structures present in an actual experiment is often in the billions. Offseting this problem is the fact that many of the DNA structures can be considered essentially identical, but only within the context of a particular experiment. That is, the criteria under which structures may be considered to be identical are dependent upon the particular experiment. For example, topology and lengths of segment are most important in the ligation experiment mentioned above and precise nucleotide sequences interior to the DNA chains are of little significance. In other experiments, dominant features involve the locations of nicks and gaps. During the simulation, a structure must be "instantiated" from a description in the pool of structures to a level of detail consistent with the intent of the experiment. Then the enzyme action is carried out on the structure resulting perhaps in several changes. Finally, the resulting structure must be reincorportated into the pool. If it is "equivalent" to another structure, then it is a simple matter to increase the appropriate concentration. Otherwise, a new structure must be added to the pool. The idea is to pick "equality criteria" and "instantiation details" broad enough to keep computations reasonable but narrow enough so that the results of the simulation correspond to laboratory results.

A second problem in simulation is the handling of impure enzymes, as for example, an exonuclease with endonuclease impurities. This may involve the construction of an event queue type of simulation in which the minor enzymatic action occurs as often as the relative concentrations indicate.

Finally, a difficulty occurs when not only qualitatively accurate answers are required from a simulation program, but also precise values of DNA structure concentrations at any moment in experimental real time. This means careful checking, probably by our geneticist consultants in the laboratory, of all contradictory rate constants, as well as possibly adding a level of mathematical rigor to some already designed models of physical processes, e.g. probability of DNA ends in a test tube solution coming close enough to join. Again, we wish to emphasize the human engineering aspects we intend to build into all of our programs and probably we can rely on our experience with DENDRAL and MYCIN. Full facilities for examining intermediate results in a natural manner to geneticists will be provided, as will powerful interactive methods of control. The user will be able to easily modify rate constants, starting DNA concentrations, and physical properties like temperature and pH during the simulation, and he will be able to trace a process backwards and restart from any point with new parameters. The simulator will interact with the DNA structure editor to allow facile entry, modification, and display of all structures.

3.4 Knowledge Base

The knowledge which must be represented in a problem solving system can be classified into three major categories:

1. knowledge which can be computed using a formal algorithm
2. knowledge (rules or procedures) for which no well-defined algorithm exists but for which good heuristics (based on expertise in the field) exist or can be developed.
3. factual data

A strong attempt will be made to represent knowledge in a uniform manner. Every item in the base could be viewed by the system in terms of a transformation at some level of detail. Some transformations combine, separate or modify substances, some merely increase knowledge. A planning program could view all data in this manner. Certainly much of the knowledge in the first two categories can be represented by procedures or rules, while many different data structures will be used for the representation of factual data. Some of the factual data may be incorporated into an algorithm or heuristic procedure. The knowledge base will be organized in a hierarchical manner so that it is easy for the system to access specific subclasses of information, such as enzyme knowledge, specific experimental techniques, or DNA structural data.

Central to the design of the knowledge base will be ensuring that data entry and modification by the expert geneticist is done in a way natural to him. This means providing a descriptive language which allows the geneticist to express the diverse types of knowledge in a language that is appropriate to the problem domain. The MYCIN system offers an excellent example to follow. It translates the input of the expert to an internal representation and then gives the expert a paraphrase of the input. The expert can correct the paraphrase interactively until he is satisfied that the program has understood correctly. With the diversity of knowledge MOLGEN is intended to handle, we may ultimately have several different language subsets for specialized use.

It is particularly important in a rapidly growing field such as molecular genetics, that the knowledge base be easy to modify and expand. Again, the MYCIN example is an excellent one. Any user can add new rules to his own working space. If these rules prove useful, the system staff adds them to the MYCIN program.

A difficult, important problem is the checking for internal consistency of the knowledge base. Eventually, we hope to develop methods to check the internal consistency of subsets of the knowledge base. For example, inconsistency in the enzyme descriptions could cause application errors which would appear as incorrect planning steps. Checking for consistency of the enzyme subset of the knowledge base could alleviate this problem.

Another feature of our knowledge base will be a literature reference or other source identification for each item represented. This source documentation will be referred to by the explanation system and will also be directly retrievable.

All of the design criteria outlined for the knowledge base in general apply to the enzyme knowledge. It can be ordered hierarchically: by enzyme function, initial substrate, product substrate, pH levels. There is knowledge that fits into each of the three general categories mentioned above. Furthermore, the type of information needed for each enzyme is similar: name, reference, basic type, substrate description, reaction catalyzed, and modifying information about parameters such as pH, salt concentration and inhibitors.

We expect the design of our enzyme knowledge base to be a dynamic process lasting at least a year. The description language will surely change as geneticists attempt to supply information using it. Building a reasonably complete file for basic experiments will take time and effort for both computer scientists and geneticists.

An example of how an enzymatic description might be used by the simulation and planning programs would serve to clarify the need for comprehensive data. Ligase, and its simple function of "sealing" a nick in the DNA backbone by making a single phosphodiester bond, has been briefly mentioned previously. A straightforward simulation problem would be to determine relative populations of circular and linear DNA after given periods of time of application of ligase to known DNA structures. For this simulation to be accurate, precise rate constants of ligase action, and how they are affected by conditions like pH, salt concentration, temperature, etc. must be provided in the enzyme knowledge base. In general, the simulator will be accessing the chemical details of the enzymatic mechanism. The planning program, however, requires more information on applicability of enzymes to the problem being considered--what substrate will a given enzyme act on, what types of DNA will compete with, or inhibit the desired enzymatic action. For example, if the geneticist wished a plan for inserting a segment of foreign DNA into a host molecule for replication, the planning program would have to pick an appropriate ligase from a selection of possible candidates. Discriminating factors would be those just discussed, substrates and inhibitors, as well as how well experimental conditions would fit in with the rest of the plan. To summarize, the simulator needs "acting" information; the planner requires "discriminating" information.

The organization of the knowledge base is central to the design of the system. The enzyme knowledge base will be used to test the ideas sketched here. Of course, we will need to add other types of knowledge concerning heuristics for planning, information about laboratory techniques and physical processes in order to have a workable system.

3.5 A DNA structure entry and editing system

One of the basic routines proposed for the molecular genetics program is an editor for DNA (EDNA), already partially completed. The idea is to have an interactive routine which accepts "text editor"-style commands allowing easy manipulation of DNA structures which are presented to the user in pictorial form. The inspiration for such an editor is drawn from an analogous routine in the DENDRAL project which facilitates the viewing and manipulation of chemical structures. The creation of the chemical structure editor has brought the internal representations of chemical structures out to the expert chemist user in a form that is natural to him and easy to use. The result has been a tremendously increased use of DENDRAL by chemists and an immediate incorporation of the tool by other programmers working on various parts of the project. We expect the EDNA routine to be used as a basic tool in many programs within the molecular genetics project.

In its completed form, EDNA will provide the user with the ability to edit DNA structures, build large structures from smaller ones, view them with several optional levels of detail, and save them on file. In many cases, structures and parts of structures will be referenced by name. It would be a simple matter, for example, for a user to read a "T6-phage" DNA structure from a file and print out its genetic map or any other level of detail to the extent that it is known by the system. New details could be entered using easy "insert segment" or "edit segment" commands. EDNA would be called by other programs, for example, by the simulator. The simulator will call EDNA so that the user can specify the initial DNA mixtures and again to print out the results of the simulation or in explaining the actions on structures.

Underlying the pictorial representations created by EDNA is an internal list structure representation of DNA. For example, a nucleotide is represented by a node which contains information to distinguish between DNA and RNA, the pyrimidine and purine bases, as well as their methylated derivatives. The node includes "3′", "5′", and "H" pointers to other nodes in the structure representation corresponding to the naturally occurring chemical bonds of the same names. Nicks and gaps in the DNA can be represented implicitly in the list structure. Other formalized types of nodes are used to represent sections of DNA where the information is less complete, that is, where the bases or the exact locations of particular features are not known.

The EDNA program is already partially written and tested. At this time various routines for drawing structures at different levels of detail are running as are the basic routines for manipulating the nodes in the list space. Several trial structures have been drawn and saved on files including some structures with hairpin configurations and others involving nicks and gaps. The structure editing commands are currently being implemented and the methods for superimposing higher biological orders of structure, for example, the superstructures of genes and special codons, are still in the design stage.

4 Resources

The principal computer science personnel involved in the design and construction of the system components described in part III of this proposal will be Professor Nancy Martin at the University of New Mexico, and two computer science doctoral thesis students at Stanford University, Peter Friedland and Mark Stefik. Molecular genetics knowledge, expertise, insights, techniques, and experimental heuristics will be provided by the researchers in Professor Joshua Lederberg's laboratory at Stanford, particularly post-doctoral fellow Stanislav Ehrlich, and graduate student Jerry Feitelson. Professor Lederberg himself will provide substantial amounts of time on a regular basis for directing the project from the genetics viewpoint. Professor Edward Feigenbaum and Dr. Bruce Buchanan will direct the computer science aspects of the project.

Offices for the MOLGEN project will be provided within the Stanford Heuristic Programming Project so as to foster interaction and exchange of ideas with workers on similar projects. Active projects within the Heuristic Programming Project include DENDRAL, a knowledge-based system for the analysis of organic compounds from spectroscopic data, MYCIN, a system for the diagnosis and treatment of infectious disease, and a project for the determination of protein structures from x-ray diffraction data. Approximately thirty workers including faculty, research associates, and graduate students are involved among the projects. All of these projects are active in the design of intelligent systems for specific application areas and there has been considerable benefit from exchange and comparison of ideas.

The superb computing facilities of the NIH-supported SUMEX-AIM timesharing installation (Carhart 1975) will be available at no charge to this project. The SUMEX-AIM facility, with Prof. Lederberg as principal investigator, is a national resource for the application of artificial intelligence techniques to problems in biology and medicine. Resources to be provided will include all CPU-time and storage required. Those involved at Stanford will be operating through hard-wired or dial-up equipment to the SUMEX PDP-10, while those at the University of New Mexico will access the system through either the ARPA network or TYMNET.

The SUMEX-AIM facility is a powerful interactive computing system open to a national community. SAIL (Stanford Artificial Intelligence Language) and other high level languages are available and supported by a large system staff. Many convenient text editors for developing programs are provided. The TENEX operating system supports flexible file handling and sophisticated storage management for a highly interactive computing environment.

5 Bibliography

Bates, D. J. and Frieden, C., 1973. "A Small Computer System for the Routine Analysis of Enzyme Kinetic Mechanisms," Comp. and Biomed. Res., 6, pp. 474-486.

Bertazzoni, U., Ehrlich, S. D., and Bernardi, G., 1973. "Radioactive Labeling and Analysis of 3′-terminal Nucleotides of DNA Fragments," Biochimica et Biophysica Acta, 312, pp. 192-201.

Bloomfield, V. A., Crothers, D. M., and Tinoco, I., Jr., 1974. Physical Chemistry of Nucleic Acids, Harper and Row.

Buchanan, B. G., 1975, "Applications of Artificial Intelligence to Scientific Reasoning,", 2nd USA-JAPAN Comp. Conf. Proc., pp. 189-194

Buchanan, B. G., Sutherland, G. L., and Feigenbaum, E. A., 1969. "Heuristic DENDRAL: A Program for Generating Exploratory Hypotheses in Organic Chemistry," Machine Intelligence 4, pp. 121-157.

Carhart, R. E., Johnson, S. M., Smith, D. H., Buchanan, B. G., Dromey, R. G., and Lederberg, J., 1975, "Networking and a Collaborative Research Community: A Case Study Using the DENDRAL Programs," to appear in Computing Networking in Chemistry, Peter Lykos, ed., American Chemical Society Symposium Series, No. 19, 1975.

Chance, E. M., 1967. "A Computer Simulation of Oxidate Phosphorylation," Comp. and Biomed. Res., 1, pp. 251-264.

Chance, E. M. and Shephard, E. P., 1969. "Automatic Techniques in Enzyme Simulation," Comp. and Biomed. Res., 2, pp. 321-328.

Corey, E. J. and Wipke, W. T., 1969. "Computer-assisted Organic Synthesis," Science, 166, pp. 179-191.

Crothers, D. M., 1968. "Melting Curves for DNA," Biopolymers 6, pp. 1391-1404.

Crothers, D. M., 1971. "Theory of the Influence of Oligonucleotide Chain Conformation on Double Helix Stability," Biopolymers, 10, pp. 1809-1827.

Davis, R., Buchanan, B. G., Shortliffe, E. H., 1975, "Production Rules as a Representation for a Knowledge-Based Consultation Program,", Computer Science Department Report No. STAN-CS-75-519

Dugaiczyk, A., Boyer, H. W., Goodman, H. M., 1975. "Ligation of Eco Rl Endonuclease-generated DNA Fragments into Linear and Circular Structures," J. Mol. Bio., 96, pp. 171-184.

Ehrlich, S. D., Torti, G., and Bernardi, G., 1971. "Studies on Acid Deoxyribonuclease. IX. 5′-hydroxy-terminal and Penultimate Nucleotides of Oligonucleotides Obtained from Cal Thymus Deoxyribonucleic Acid," Biochemistry, 10, 2000-2009.

Elton, R. A., 1974. "Theoretical Models for Heterogeneity of Base Composition in DNA," J. Theo. Bio., 45, pp. 533-553.

Fikes, R. E., Hart, P. E., and Nilsson, N. J., 1972. "Some New Directions in Robot Problem Solving," in Machine Intelligence 7, Edinburgh University Press, pp. 405-430.

Garfinkel, D., 1968. "A Machine-independent Language for the Simulation of Complex Chemical and Biochemical Systems," Comp. and Biomed. Res., 2, pp. 31-44.

Green, S. B. and Garfinkel, D., 1970. "Simulation of Enzyme Systems Using a Matrix Representation," Comp. and Biomed. Res., 3, pp. 166-173.

Harris-Warrick, R. M., Ehrlich, S. D., Elkana, Y., and Lederberg, J., 1975. "Reaction and Purification of Bacterial Genes by Segmentation of DNA with Eco R1 Endonuclease and Agarose-gel Electrophoresis," Proc. Nat. Acad. Sci., USA, August 1975, pp. 2207-2211.

Hart, P., Nilsson, N., Raphael, B., 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Sys. Sci. Cybernetics, Vol. SSC-4, 2, pp. 100-107.

Kelly, T. J., Jr. and Smith, H. O., 1970. "A Restriction Enzyme from Hemophilus Influenzae. II. Base Sequene of the Recognition Site," J. Mol. Bio., 51, pp. 393-409.

Kulikowski C A, Weiss S, Saifr A., 1973. "Glacoma Diagnosis and Therapy by Computer," Proceedings of Annual Meeting of Ass. for Reserch in Vision and Opthamology.

Nathans, D. and Smith, H. O., 1975. "Restriction Endonucleases in the Analysis and Restructuring of DNA Molecules," Ann. Rev. of Biochem., 44, pp. 273-193.

Nilsson, N. J., 1971. Problem Solving Methods in Artificial Intelligence, McGraw-Hill.

Pople, H. E., Myers, J. D., and Miller, R. A., 1975. "DIALOG: A Model of Diagnostic Logic for Internal Medicine," Fourth Int. Joint Conf. on Art. Intel., 2, pp. 848-855

Powers, G. J., Jones, R. L., Randall, G. A., Caruthers, M. H., van de Sande, J. H., Khorana, H. G., 1975. "Optimal Strategies for the Chemical and Enzymatic Synthesis of Bihelical Deoxyribonucleic Acids," J. Am. Chem. Soc., 97, pp. 875-888.

Rau, D. and Klotz, L. C., 1975. "A more Complete Theory of DNA Renaturation," J. Chem. Phys., 62, pp. 2354-2365.

Reddy, D. R., Erman, L. D., and Neely, R. B., 1973, "A Model and a System for Machine Recognition of Speech". IEEE Transactions on Audio and Electroacoustics, AU-21, p229.

Sacerdoti, E. D., 1973. "Planning in a Hierarchy of Abstraction Spaces," Third Int. Joint Conf. on Art. Intel., pp. 412-422.

Shortliffe, E. H., Axline, S. G., Buchanan, B. G., Merigan, T. C., and Cohen, S. N., 1973. "An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Theory," Comp. and Biomed. Res., 6, pp. 544-560.

Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C., and Cohen, S. N., 1975. "Computer-based Consultations in Clinical Therapeutics: Explanation and Rule-acquisition Capabilities of the MYCIN System," Comp. and Biomed. Res., 8, 303-320.

Siklossy, L. and Dreussi, J., 1973. "An Efficient Robot Planner which Generates its own Procedures," Third Int. Joint Conf. on Art. Intel., pp. 423-430.

Sklar, J., Yot, P., and Weissman, S. M., 1975. "Determination of Genes, Restriction Sites, and DNA Sequences Surrounding the 6S RNA Template of Bacteriophage Lambda," Proc. Nat. Acad. Sci, USA, 72, pp. 1817-1821.

Smith, D. H., Buchanan, B. G., Engelmore, R. S., Aldercreutz, H., and Djerassi, C., 1973, "Applications of Artificial Intelligence for Chemical Inference IX. Analysis of Mixtures without Prior Separation as Illustrated for Estrogens," J. Am. Chem. Soc., 95, 6078

Sobell, H. M., 1973, "Symmetry in Protein-Nucleic Acid Interaction Advances in Genetics, Academic Press, pp 411-490

Sulkowski, E. and Laskowski, M., Sr., 1962. "Mechanism of Action of Micrococcal Nuclease on DNA," J. Bio. Chem., 237, pp. 2620-2625.

Winograd, T., 1972. Understanding Natural Language, Academic Press.

Wipke, W. T., Gund, P., and Friedland, P., 1975. "ALCHEM: A Language for Describing Chemical Transformations," in preparation.

Wong, A. K. C., Reichert, T. A., Cohen, D. N., and Aygun B. O., 1974. "A Generalized Method for Matching Informational Macromolecular Code Sequences," Comp. in Bio. and Med., 4, pp. 43-57.

7. BUDGET

NATIONAL SCIENCE FOUNDATION
Washington, D. C. 20550

RESEARCH GRANT PROPOSAL BUDGET (TWO YEAR TOTAL)
2 Year Beginning 6/1/76

Institution: Stanford University
Principal Investigator(s): E. A. Feigenbaum, J. Lederberg
Program Name: MOLGEN: A Computer Science Application to Molecular Genetics

	NSF Funded Man-months			Proposed Amount
	Cal	Acad	Sum	
A. SALARIES AND WAGES:				
1. Senior personnel:				
a. (Co) Principal Investigator J. Lederberg***	-	-	-	-
(list by name) . . E. A. Feigenbaum		18 (10%)	2 (100%)	11,954
b. Faculty Associates				
(list by name)				
(Sub-total)				
2. Other personnel (Non-faculty)				
a. Research Assoc. (Post-doctoral) (list separately by name if available, otherwise give numbers)				
Bruce G. Buchanan, Research Computer Scientist	11(25%)			6,838
b. Non-Fac. Professionals (Other) (list separately--by category, giving number, e.g. one computer programmer)				
. . . .				
. . . .				
. . . .				
c. (3) Grad Students (Res. Asst.)				32,478
d. ( ) Pre-Baccalaureate Students				
e. ( ) Secretarial-Clerical				
f. ( ) Technical, Shop & Other				
Total Salaries and Wages				51,270
B. STAFF BENEFITS:				9,711
C. TOTAL SALARIES, WAGES AND STAFF BENEFITS (A + B)				60,981
D. PERMANENT EQUIPMENT: (List as Required)				
. . Purchase of two computer terminals				5,180
Total Permanent Equipment				5,180
E. EXPENDABLE SUPPLIES AND EQUIPMENT				1,000
F. TRAVEL:				
1. Domestic				2,000
2. Foreign (list as required)				
Total Travel				2,000
G. PUBLICATION COSTS				400
H. COMPUTER COSTS (if charged as direct costs)				
I. OTHER COSTS:				
(itemize by major type) Terminal Maintenance				960
Communications (terminal-to-computer, project business phone, postage)				1,500
Total Other Costs				2,460
J. TOTAL DIRECT COSTS (C through I)				72,021
K. INDIRECT COSTS: +				
1. On Campus . . . . .% of				41,523
2. Off Campus . . . . .% of				
Total Indirect Costs				41,523
L. TOTAL COSTS (J plus K)				113,544
M. TOTAL CONTRIBUTIONS FROM OTHER SOURCES				
N. TOTAL ESTIMATED PROJECT COST				113,544

NATIONAL SCIENCE FOUNDATION
Washington, D. C. 20550

RESEARCH GRANT PROPOSAL BUDGET
Year Beginning 6/1/76

Institution: Stanford University
Principal Investigator(s): E. A. Feigenbaum, J. Lederberg
Program Name: MOLGEN: A Computer Science Application To Molecular Genetics

	NSF Funded Man-months			Proposed Amount
	Cal	Acad	Sum	
A. SALARIES AND WAGES:				
1. Senior personnel:				
a. (Co) Principal Investigator J. Lederberg***	-	-	-	-
(list by name) .....E. A. Feigenbaum..(10%)...		9		2,753
b. Faculty Associates				
(list by name)				
(Sub-total)				
2. Other personnel (Non-faculty)				
a. Research Assoc. (Post-doctoral) (list separately by name if available, otherwise give numbers)				
....				
b. Non-Fac. Professionals (Other) (list separately--by category, giving number, e.g. one computer programmer)				
....				
....				
....				
c. (3) Grad Students (Res. Asst.)				16,224
d. ( ) Pre-Baccalaureate Students				
e. ( ) Secretarial-Clerical				
f. ( ) Technical, Shop & Other				
Total Salaries and Wages				18,977
B. STAFF BENEFITS:				3,409
C. TOTAL SALARIES, WAGES AND STAFF BENEFITS (A + B)				22,386
D. PERMANENT EQUIPMENT: (List as Required)				
..Purchase of two computer terminals**.................				5,180
Total Permanent Equipment				5,180
E. EXPENDABLE SUPPLIES AND EQUIPMENT				500
F. TRAVEL:				
1. Domestic				1,000
2. Foreign (list as required)				
Total Travel				1,000
G. PUBLICATION COSTS				200
H. COMPUTER COSTS (if charged as direct costs)				
I. OTHER COSTS:				
(itemize by major type) Maintenance of computer terminals				480
Communications (terminal-to-computer, project business phone, postage)				750
Total Other Costs				1,230
J. TOTAL DIRECT COSTS (C through I)				30,496
K. INDIRECT COSTS: +				
1. On Campus .....% of				17,438
2. Off Campus .....% of				
Total Indirect Costs				17,438
L. TOTAL COSTS (J plus K)				47,934
M. TOTAL CONTRIBUTIONS FROM OTHER SOURCES				
N. TOTAL ESTIMATED PROJECT COST				47,934

NATIONAL SCIENCE FOUNDATION
Washington, D. C. 20550

RESEARCH GRANT PROPOSAL BUDGET
Year Beginning 6/1/77

Institution: Stanford University
Principal Investigator(s): E. A. Feigenbaum, J. Lederberg
Program Name: MOLGEN: A Computer Science Application to Molecular Genetics

	NSF Funded Man-months			Proposed Amount
	Cal	Acad	Sum	
A. SALARIES AND WAGES:				
1. Senior personnel:				
a. (Co) Principal Investigator J. Lederberg***	-	-	-	-
(list by name) E. A. Feigenbaum		9 (10%)	2* (100%)	9,201
b. Faculty Associates				
(list by name)				
(Sub-total)				
2. Other personnel (Non-faculty)				
a. Research Assoc. (Post-doctoral) (list separately by name if available, otherwise give numbers)				
Bruce G. Buchanan, Research Computer Scientist	11 (25%)			6,838
b. Non-Fac. Professionals (Other) (list separately--by category, giving number, e.g. one computer programmer)				
c. (3) Grad Students (Res. Asst.)				16,254
d. ( ) Pre-Baccalaureate Students				
e. ( ) Secretarial-Clerical				
f. ( ) Technical, Shop & Other				
Total Salaries and Wages				32,293
B. STAFF BENEFITS:				6,302
C. TOTAL SALARIES, WAGES AND STAFF BENEFITS (A + B)				38,595
D. PERMANENT EQUIPMENT: (List as Required)				
Total Permanent Equipment				
E. EXPENDABLE SUPPLIES AND EQUIPMENT				500
F. TRAVEL:				
1. Domestic				1,000
2. Foreign (list as required)				
Total Travel				1,000
G. PUBLICATION COSTS				200
H. COMPUTER COSTS (if charged as direct costs)				
I. OTHER COSTS:				
(itemize by major type) Terminal maintenance				480
Communications (terminal-to-computer, project business phone, postage)				750
Total Other Costs				1,230
J. TOTAL DIRECT COSTS (C through I)				41,525
K. INDIRECT COSTS: +				
1. On Campus .....% of				24,085
2. Off Campus .....% of				
Total Indirect Costs				24,085
L. TOTAL COSTS (J plus K)				65,610
M. TOTAL CONTRIBUTIONS FROM OTHER SOURCES				
N. TOTAL ESTIMATED PROJECT COST				65,610

BUDGET NOTES

Salary increases estimated at 10%, effective Sept. 1.

* Equal to 2/9 academic year salary.
** Over two-year period, lease price exceeds purchase price plus maintenance. However, leases can be arranged if administratively more convenient to NSF.
***Professor Lederberg's activity on this project will be done without charge to the budget.

+ INDIRECT COSTS: On Campus
56% of Total Direct Costs thru 9/1/76
58% of Total Direct Costs thereafter.