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., Droney, 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 R1 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 Bioned. Res., 2, pp. 31-44.

Green, S. B. and Garfinkel, D., 1970. "Simulation of Enzyme Systems Using a Matrix Representation," Comp. and Bioned. 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 Bioned. 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 Bioned. 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 Cal	Acad	Sum	Proposed Amount
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 Cal	Acad	Sum	Proposed Amount
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 Cal	Acad	Sum	Proposed Amount
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.