Lederberg / Lindsay 1-9-75

RL: So, with that background, if you could tell me what you see as the major ideas and the major contributions which have come forth from the DENDRAL project.

JL: Well, I'm an ignoramous about Computer Science as a discipline. I spend a lot of time trying to learn things about it and I do as I go along, but in a certain sense I'm not well-equipped to answer that question because I don't know the myths, the intellectual structure of the field and so on well enough to know in what way it was very obvious, that the DENDRAL Project what was going on there which made ... and so on. I've heard comments about that from time to time from Ed and Bruce. And, in fact, the main thing that I would say that I've learned in this direction is that a job like this can be done. That it is possible to engineer a system of this kind. I find it hard to lay down what are the broad theoretical accomplishments; I haven't been that self-conscious about the theoretical structure of what it was that we were doing. And that's a point about which I certainly can be acutely self-critical--it's not quite the way I would approach things in my own discipline, I am a little skeptical about the extent to which any enormous theoretical structure has developed as a result of this effort. I think we've done a very good job of engineering and to try to figure out wherein what we did from point to point differed greatly from common sense and experimental probing about a few things that work and a few things that don't work, and so on, and packaging it in an intelligent and orderly fashion. I'm a little at a loss to describe, partly I suppose because I don't have the theoretical framework which makes such a description. That may be a more modest view of what we've been up to than my friends would be willing to admit to. It's not something I'm insisting on, I'm just telling you. I did outline a couple of things in a note that came out a few years ago, a status report. Let me see if I can find that particular one because isolating things out of the DENDRAL reports...

RL: I don't think I've seen that particular one.

JL: You probably have. It's under some other heading or name.... I couldn't find it in that pile but I'm sure I'll find it here. Yeah, this is the one. So it's report No. 104 and it's not exactly the same as the way it appeared in Machine Intelligence V so that's... Well, let me ponder on that a little bit. I guess I started out with a lot of prejudices about the design of the system without having had necessarily very much of a theoretical framework about what other people were doing, and so I may indeed have had a fairly strong explicit theory in mind without knowing that I did speak in prose for a long time. The notion of a canonical generator is one that I guess I've not really seen expressed explicitly anywhere else, but I sort of live with it as a given from which one then goes into heuristic pruning exercises. And that's in a nutshell what DENDRAL does, and the issues of how you go about doing it and tuning it to the reality of the situation -- so maybe that deserves some emphasis: in trying to discriminate problems for their amenability to this approach, that canonical generator looms very, very large, and particularly with the criteria for equivalence, and I have frequently told myself I'd be willing to go into any other scientific field, I would like to try to break out of chemistry, if I could satisfy the criterion of having a notation in which hypotheses could be expressed, and a machine that could test statements for semantic equivalence to one another. Now we can do that for structures of organic molecules and that is just about all as far as real world oriented science is concerned. A lot of mathematics, obviously, one can make transformations of expressions ... have that property.

RL: Would it be fair to say that if you found a field in which there were no structural generator that DENDRAL has no lessons in that area?

JL: No, I think that there has been a good deal of shrewdness in solving many other kinds of problems down the way that don't require the generator, and so that's too strong a statement. I think the central concept of DENDRAL is built around that approach to the selection of hypotheses and there really is an exhaustive generator that can construct valid statments; even before you look at the data you have some way of parsing through all acceptable sentences and being sure that you had all of them, and so on.

RL: Do you have any advice for somebody as to how to go about discovering or inventing such a generator in a different area?

JL: No, not especially. Well, I don't know whether I do or not. Again, I have an image of what to do about it in chemistry and one is able to map hypotheses of analytical organic chemistry onto some fairly elementary algebraic concepts, graphs, and one knows the properties of automorphism and from that you can generate the generator; in fact it took a lot of fairly particular hacking away at it to discover efficient ways of building the generator -- it's one thing to say let us produce all possible non-equivalent representations and another to do it in a way that does not involve an enormous amount of back comparisons, of weeding out of redundancy explicit search for equivalences, and that sort of thing. These go under the heading roughly of labelling problems and when we come to the cyclic graphs the situation is not quite so straightforward and it took quite a while to get a good way of handling that -- it just came out fairly recently. So I'm not sure that one is in a position to generalize from that about how you go about doing it in another field, we ought to re-examine that question. That's a very interesting question; I've never looked at it quite that way. All I can say is that the first step is to look for procedure that would do that, is to try to find some way of organizing the generator that is prospectively efficient, that has in it built-in constraints, so that you can guarantee and demonstrate in some reasonably rigorous fashion that it has the properties that you have been describing. For trees, that was quite straightforward; for rings, it was somewhat more complex; but the first question, the one that I think is a little bit of an innovation to many people, is knowing that that's what you want to do. So I think a description of what you mean by a prospectively efficient generator could be a very important element in any books of this kind. Now there's some fields where that simply doesn't apply. If you're talking about chess, your move generator... there's such a total lack of symmetry in the game that..., and then positions at each move are relevant, you're not only interested in final states, you have to match your situations move by move; you can't go through a transition that involves a check-mate and have something the other side of it, you see, which is a little different from some of the generations that we go into. So there the total lack of symmetry in effect gives you no, you might say, either no difficulty, or no opportunity, but there's no weeding, the algorithm that I can think of that would reduce the combinatorial space of valid moves. But maybe that's not completely the... to a first approximation that's true; however, you can't move your king across a file that's controlled by a queen and things of that sort, so if one stopped to think about it, maybe there's some minor exceptions. But it's obvious that the exceptions don't dominate the situation. Well in the case of organic chemistry, if you start thinking about all possible ways of putting atoms together, the redundancies would soon swamp you out if you didn't have some rules to take care of the symmetries that you do run into, so different fields will have different rules for this. In natural language I would think that isosemanticism would be a difficult and fantastic problem, that is, there are so many different ways of saying the same thing, that I have prospectively despaired of even going into it until that particular art has gone very much further and so I have not even thought very much about trying to DENDRALize areas of science in which one could not develop a reasonably formal notation other than a natural language. I really see no hope of using natural language.

RL: Can you think of any other scientific fields, either in your own specialty, genetics, or related subjects like organic chemistry, in which there are projects which could be attacked and better progress be made now that you've done DENDRAL? That is, are there any lessons that could be extended to other similar fields?

JL: Well, I've already indicated some despair about the generator side of it, although that's not total despair. We're taking a look now at two different aspects of genetics to see how we might encode them. One of them is essentially mathematics, so it can almost ipso facto be translatable; it's a very theoretical branch of genetics, population genetics. And, I don't know how familiar you are with it, for the elementary level, but the sort of problem that I would pose is, How do you build a machine that could discover the Hardy-Weinberg Law, which is a fairly simple combinatorial property of the results of a random mating within a population. And there you see the system is so completely, is already formal, you can express everything that you're interested in in that field in algebra, and so there really are no difficulties in representation. What to do about a generator there, what are not only valid but interesting statements, becomes the next horizon on that. And I think it's something worth looking at, but I haven't actually done it yet. Jon King is beginning to look at it. He did a class project this last term which I still have to evaluate and criticize, but you might want to talk to him and see what ideas he's developed. Another area that is in a way/akin to organic chemistry and in fact is exactly where we're going to be able to branch out most readily is in the molecular genetics, in the behaviorax of D and A molecules. And there I would say that we're not dealing with anything that's fundamentally different from the structures that we're dealing with in organic chemistry, but the representations are altered slightly. We'll be talking about strings of D and A molecules in various kinds of associations in D and A sequences, rather than individual atoms with the simple connectivities that we've had before. But otherwise I think the basic notions are not altogether different. We have a fairly definite number of rules and about how the D and A molecules behave with respect to one another; how they hyberdize; what enzymes will attack them; what kinds of pieces are left after enzymatic treatment and statistical descriptions about what happens when D and A is broken, and things of that sort. So there we have a series of mechanisms that are sufficiently close to what we have in organic chemistry without being quite identical to them. I think we do have a chance to try to formalize that a bit. And the sort of problems we run into there are mechanizing the kind of imagination that suggests new sorts of experiments to do. And we would then need a formal language to describe those experiments. It seems graspable and I think it's something we should be able to get on to with much less effort than the first round of DENDRAL. The other areas that one might contemplate I think do suffer from the formal statement problem and I think there are probably quite a number of fields where that can be done, but it's a formidable effort. It's not one I know as much about as I'd like to. Pat Suppes might have something to say about that. We've had a few conversations on this point. There are a few formal systems in psychology, sociology, Woodyer tried to do something in embryology, a few years ago. In fact, Woodyer also did a piece on psychology in which he quite literally tries to express a number of concepts, building them up from propositional calculus and avoiding natural language totally. I'm not really in good position to judge those; I gather they haven't made very much impact on the field that they were in, except as first trials of trying to do it. I think that's where we're at right now; I don't think there's been a very great effort to attempt to formalize them. People outside of mathematical logic I think sometimes try to develop formal systems, and people inside the field have despaired and gone to less and less formal representations of what they're doing. But I think without the motivation of putting them into computer programs you won't have the sense of need to do that step work that's necessary to do that kind of translation. Some people think the way around that is to wait for the natural language hackers to get far enough along that you can just give them our own natural language text and programs extract them ... I think it will be a very long wait.

Well, that's one piece of it. There are a few perceptions, strategy that are mentioned in that article, but they really have much more to do with engineering tidiness and avoiding some fairly obvious traps that are obvious after you've been in them, than any great theoretical doctrine. And maintaining the logical consistency of your system is really much more difficult than you would ever believe. As you keep maintaining it and correcting a little piece of it you're just constantly knocking other things down that you weren't aware of at the time you were laying it all out. And so your notion of putting your basic system of rules and axioms legitimate rules and so forth, in one place and making sure that the program generates all the code that it needs throughout the system, from one consistent source, I think is a lesson you learn the hard way. It still hasn't been done completely perfectly and systematically, but wherever possible it is, and we've had a much happier time of it since then. A great theoretical contribution or just a rule of experience. But if the first time you ever tried to do something like this, if you're not aware of that, and it hasn't been knocked into you, you can flounder around for a very long time.

It ends up being very similar though to the general programming problem, which I don't think is all that different from artificial intelligence. Very complex algorithms and how to keep them running. How to maintain them and keep them running well is part and parcel of the problem of AI. I don't really see a very sharp boundary between AI and other complicated algorithms. The other kinds of things that can be done in looking for shortcuts is not only rely on the real world, but you can also, once you've got a generator, that it can generate its own problem situations, then start developing you heuristics for shortcutting into them letting the generator use its sets of rules. For example, well, it would be a little bit analogous to saying you don't have to wait for all the games of the grand masters, if you're doing this in a chess program. Let the system play some of its own games. And play the problems that it itself generates in looking for the strategies. In the case of chess you may have enough material to work with. That isn't a problem. In our situation we did run out of, we would have difficulties putting in thousands of examples of known solutions to known problems. And while we put in as many as possible, in sharpening the tools for looking for the shortcuts from data to the hypotheses, since everything depends on the consistency of your generator anyhow, you might as well let it spin those out and invent data that you then use in to inverse fashion in looking for short cuts. That's something we have not really implemented to any great degree. It's been used a few times and it's successful, been on the shelf for a while. Some of the other strategies that we've developed are also, maybe there's a lessoo to be derived/ from the fact that we have ideas that haven't really beens thoroughly worked out. It's only a little while that we've had the luxury of this stable computing environment and the resources to really do the things we want to do. We're too busy to do a list of priorities of things to clean up. But the role of the dictionary is a very interesting question and strategies for using it I think have sort of the next level of A.I. and I think there's some generality on it. You know what issue I'm referring to? And we never really did address what the heuristics ought to be, how you go about making choices as to when to consult the dictionary and when not. I thing that's a rather interesting horizon to try to get into. There were a number of occasions that various people thought that had radically different approaches to the problem, they ended up to be quite mapable onto the original notions of graph generation and verious kinds of re defintions of the canons of order.

RL: What were some of those?

JL: Well, this (planner) idea. That was used... There was a specific strategy/set that was up for the amines, which doesn't look at all like the DENDRAL generator. But then if you look twice at it, you discover that it really is, only you've redefined the center of the graph, you've got some superatoms layed on, and that you could describe the entire procedure in terms of canonical generator, but just with those kinds of substitutions of arguments. We ended up discovering that DENDRAL really was a very general machine. And I find it hard to say how it could be otherwise; I mean we really are giving fundamental graphic description of the molecules and the strategies suggested really weren't totally new— they again involved graphic generations. I wasn't surprised when the tables showed that they were in fact homologous to one another. But it does say something that you really do want to write your generators in such a way that they can be internally rearranged very readily, that you don't have them locked into difficult code, that the sequence of priority of different steps can be readily altered so that you end up with a table driven approach to that, and that enables you to experiment with alternative strategies in terms of what are the most efficient ways of setting up your heuristics for different kinds of problems. That's something I've advocated mechanizing which we have not done to any appreciable degree. But some of these discoveries of new strategies involved simply in the long run were not much more than inverting the order of precedence of some of the operations in the generator, which is entirely appropriate to different situations and in which one could scan either the data or the problem space and decide if there was a more efficient strategy that could be used there.

RL: Ok, that gives me a pretty good idea of what I wanted. Do you have anything else to add?

JL: Not right off hand. I'm sure I would after some further iteration.