PS PDP BEGINNINGS ... Monday PDP models 1

-APR-85 22:15:51 holyoak thoughts on PS and PDP mod...
ubject: thoughts on PS and PDP models
ource: holyoak (keith holyoak @ cmu-psy-b)

Here is a collection of queries/comments relevant to discussion of PS and PDP models:

1) To what extent are the two systems more naturally applied to different domains of skills (e.g., PS for conscious problem solving, PDP for low-level perception)?

2) The issue of whether the distinction is one of level of description or of substance seems to demand a clear characterization of what the defining properties of each system are. In particular, it could be argued that many of the claimed virtues of PDP (graceful degradation, simultaneous constraint) can be exhibited by PS systems with parallel activity of rules (e.g., Soar, classifier systems of John Holland).

3) What are the virtues and deficits of each system with respect to theoretical constraints on representations? For example, PS needs to constrain complexity of rules (#elements, logical form, use of variables); PDP must specify initial connections that undergo strength revision.

4) An apparent difference between the two systems with respect to learning is that PS systems allow mechanisms for the generation of new rules as well as for the revision of strength of existing rules; PDP typically exhibits only revision of connection strengths. Are there empirical arguments for or against rule generation? As evidence for generalization, what about Elio & Anderson on order-of-presentation effects on category induction?

5) Can PDP systems acquire representations of higher-order relations? This seems necessary for analogical mapping in general.

6) Where does seriality of thought/behavior emerge from each system? Where do resource limitations come from?

7) People are smarter than rats (in some ways). Is it possible to point to aspects of either type os system that might account for apparent qualitative differences in intelligence?

8) Can PDP account for learning by instruction? By imitation?

9) How are weak problem-solving methods represented in PDP? How are stronger methods acquired?

10) Can PDP account for evidence there are inferential rules applicable to broad classes of domains (e.g., statistical inference, heuristics for assessing causation, Lenat's extreme-case heuristic)?

11) How do PDP and PS handle evidence for effects of learning orientation (e.g., Reber's implicit vs. explicit learning of categories)?
-APR-85 09:51:13 anderson first thoughts

ource: anderson (john anderson @ cmu-psy-a)

As requested, I am posting my initial reactions to the PSPDP discussions

1. I would like to address the assertion in the original post that a large fragment of the cognitive science community is becoming dissatisfied with productions systems. It strikes me as a strange claim. It seems to imply that there was a time when a large fragment was satisfied. This is certainly not the case. There only was a time when a large fragment was ignoring production systems.

If there is any rise in dissatisfaction it must be because for the first time the community has decided that it has to pay attention to production systems. If they have it is because they finally recognize the significance of production systems as a class--they are the only theoretical formalism that has successfully accounted for control of cognition--why the direction of thought has the generally successful course that it does throughout all the cognitive events we engage in. In every other theoretical formalism it has been a promise combined with a bunch of just-so stories. Production systems deliver on the detail of how it happens.

When I first recognized this about production systems I can recall an initial period of great dissatisfaction. Production systems seemed to me to be so rigid, so unnatural, so computerish that they could not describe what was going on in a flesh and blood head. However, the key to my eventual persuasion was the realization that a production rule was high level specification of a cognitive contingency of which the mind was capable. Over the years I worked out how that contingency could be implemented in a way compatible with everything I believed about flesh and blood heads. However, there remains something psychologically significant to the production level formulation--it is not just a description of a bunch of low level processes-- this is a point to which I will return.

It is interesting to contrast production systems with flowchart theories which specified a flow of control for a specific task but which did not evoke the same sense of discomfort. I think the reason is that the flowchart was just a description of the flow of control and not a theory. Hence, it did not imply any commitments about which we could be uncomfortable.

2. There are a number of analogies floating around for how to think of the relationship between the PDP level and the PS level. However, the one I would like to advance as not prejudging the issue is between machine language and a high-level programming language from LISP. At some level nobody doubts whether there is this machine-level PDP implementation, the empirical question is whether it is compiled from some higher level PS implementation. If it isn't we would expect there to be mental programs that do not perfectly correspond to any set of productions just as it is the case that not all possible machine language programs could be compiled from LISP programs. On the other hand if there is a reality to the PS level we should discover that there exists no PDP level system that does not implement a production system. Thus, at this abstract level the issue seems quite decidable.

There are a number of qualifications on this of course. One is that the PS level has a limited range of applicability. Nobody proposes to model what is happening on the retina in terms of PS systems for instance. Second, the reality of the PS level does not deny that there are phenomena that require

digging below the production. To understand the exact timing properties of cognition or exactly how it breaks down one will have to dig below the production level even as one needs to dig below LISP code to understand these issues in a computer. However, this fact does not deny the truth of the claim that there is a level of understanding at which the LISP code is a complete specification of the system and one does not need to look below.

3. Despite the comment above the empirical prospects of deciding PDP-only versus PS-implemented-in-PDP seem quite slim. This is because we are talking about frameworks for theories rather than specific theories. I can imagine and have found empirical phenomena that caused me to reject my current ACT theory but I just choose another instance in the PS framework. When we come to arguing the PS versus PDP matter we wind up playing very fast and loose picking any feature of any theory within the class to make our point. I think all we can really do is test theories within a framework allowing for some sociological phenomena larger than any of us decide which is the better theoretical framework.

On this score I think it is critical that we identify what the theories are that we are working on and what their testable assumptions are--i.e. how they might be falsified. This is the only way that we will be able to make real progress in developing theories within either framework. I think production systems score strongly here in making their testable assumptions explicit and we have seen relatively rapid theoretical progress. A similar thing needs to be done at the PDP level. I have yet to get a clear statement for instance on the importance of linear separability to learning which should be a key issue if any key issues exist for that class of theories.

4. I can name at least three features which seem to be rather key to many PS like theories and which seem out of the spirit of PDP models, though of course they are computationally universal and a just-so story could be cooked for anything:

1. The notion of entities like goals as abstract objects controlling cognition.

2. The importance of the existance of variables giving generality to rules of cognition.

3. The fact that learning takes place at the abstract rule level rather that at more specific connections among elements

5. Finally, I would like to take back part of my concession in (2) which is that everyone believes in PDP models. What everyone believes is that neural stuff underlies cognition, that neural stuff is highly parallel and at least somewhat distributed, that neurons excite and inhibit and this has to be at least some of the picture, and that learning sure better be happening somewhere in the nervous system. However, this is a long way from the specifics of a PDP model and the neural evidence for the specifics is weak at best. Moreover, what neural evidence there is applies to rats and salamanders about as well as it applies to humans, but to echo Holyoak's point, humans are qualitatively different in their cognitive facillities.

-APR-85 17:33:21 holyoak Sources of dissatisfaction...
ubject: Sources of dissatisfaction with PS models
ource: holyoak (keith holyoak @ cmu-psy-b)

I have a few reactions to John's note. He rightly points out that the first issue raised contained a questionable presupposition, to the effect that many in the field have become increasingly dissatisfied with PS models. The point is certainly debatable, and in fact I count myself as one who has grown more attracted to rule-based systems with time. My first encounter

3a) PS theorists have vacillated between claiming that their models are truly theories versus that they are general-purpose representational systems in which virtually any theory can be implemented. To the extent the latter is the case, they offer at best a tool, rather than a solution, to those who seek theories.

3b) PS models have emphasized sufficiency at the expense of necessity. As scientists we would all like to find models that are both necessary and sufficient to account for a set of interesting phenomena; i.e., sufficient to model the actual task, and necessary in the sense that empirical evidence or task analysis indicates that no model lacking certain specifiable properties could account for task performance. Such models are hard to come by, and there is disagreement (often implicit) in what we should settle for in the interim. PS models, which have been the mainstream of computer-simulation methodology in cognitive psychology, have typically stressed how fine it is to have a sufficient model -- one that really can perform the task, unlike the just-so variety to which John alluded. There is a perspective, however, from which sufficiency is deemed less compelling than necessity, if we really are forced to choose. Work such as that of Shepard et al. on the relationship between imagery and perception, and Kahneman & Tversky on judgment heuristics, are prominent examples of efforts that have generated excitement because they have attempted to establish the necessity of certain general processing principles, even though it is hard to point to sufficient models. Even if one were to proceed to, say, model the representativeness heuristic in terms of an explicit PS model, some people would consider it an academic exercise in the pejorative sense unless further constraints were derived in the process.

Of course, it isn't immediately obvious that PDP models provide superior accounts of representativeness (although I feel I may have set up Jay McClelland here). I think, however, they are currently perceived as being less concerned with sufficiency because of their spartan representational systems and learning mechanisms, and the fact that their proponents tend to cheerfully acknowledge there are some tasks their models can't perform at the moment.

4) Production systems have generally been rather rigidly serial in their behavior. This makes them unappealing to those who for whatever reason impute parallelism to thought at some level. And we all know what the first P in PDP stands for.

I think that is enough (or more) for the moment.

0-APR-85 11:31:06 schneider dissatisfaction with PS mo...
ubject: dissatisfaction with PS models
ource: schneider (walter schneider @ cmu-psy-a)

I asked some of my collegues as to is there an unconfortableness about the production system view of the world and here are some comments.

Yes, I think it's true and let me start a list of possible reasons.
1. what can go in a production? seems unconstrained
2. how can one test any particular proposal for a set of productions?
3. how can we tell what a production system will do? There are two parts to this (a) as with any complex model, predictions only arise from the simulation, but that means they're not equally availabe to everyone for testing and may depend on non-theoretical aspects of the production system used
(b) production system for any task is supposed to be part of a huge set of productions for all other tasks, but rarely (though sometimes) is it shown that integrating these will not lead to problems (i.e., it seems like conditions, while they seem reasonable now, would have to be a lot greater)

One more thought. For production system models of LTM (e.g., Newell), memory does not have the organized quality
that we (as psychologists) have come to believe in. Clearly, this does not apply to models where production systems are the procdeural system only.

---

another view

I THINK THE statement is true. The Production systems are an artifice of the CS community. It is not clear that the idea came from any consideration of brain function. Rather, if-then statements are used to control which processes are to occur within a given environment. The if-thens themselves are ad-hoc as is the control scheme used to determine which of the active rules to use. The good thing about production systems is that they are data-driven---if-then rules fire when the data is of the right configuration. But who thinks that the brain really has memorized rules for each situation and a control scheme for their selection? This seems more of a programming heuristic for making COMPUTATIONAL machinery do clever things.

---

some written comments from D Dennetts Aview from the East Pole
p27 "according to the Western view, the apportionment of responsibility and power between memory and intelligent processing will be unlike the underlying (and ineluctably influential) division of labor in von Neumann machines, in which actions happens in the central processing unit; in which memory is inert, cold storage and all the action happens in the central processing unit; a proper memory will do a great deal of intelligent work itself.
p32 "And Newell and Simon's search for "rules" of "thinking" need not commit them or their admires to the HCC[High Church Computationalism] doctrin that thinking is rule-governed computation. The rules they discover (suposing they succeed) may instead be interpreted as regularities in patterns in the emergent phenomena."

0-APR-85 11:58:35 klahr cognitive development
ource: klahr (david klahr @ cmu-psy-a)

My contribution to this discussion will be to provide a developmental perspective. As I see it, the fundamental theoretical questions in cognitive development are:

(1) what is the "innate kernel"? That is, what is the core collection of knowledge structures and processes at T=0?

(2) what are the self-modification processes?

One might argue that the notion that "innateness" is important is arbitrary: we could be just as interested in what's available at T=t as in what's there at T=0. That is, one might construe all learning models as asking the same question as (1) but just having a large value for T. However, most (all?) learning models face the serious problem of accounting for the initial knowledge states they attribute to the system they are dealing with. Although the models can go
from state n to n+1 via the proposed learning mechanisms, it is not clear that the same set of learning mechanisms could have gotten the system to its starting state in the first place. Thus, by posing the question in terms of innate processes, we force the issue into the open.

The answer to (2) will have to include "conventional" learning mechanisms -- which base their learning on feedback from the environment -- as well as "reflective" or "self-contained" mechanisms, which can effect changes in the system based only on internally detected regularities, inefficiencies, redundancies, and so on. Additionally, these mechanisms will have to function spontaneously, without the establishment of any special learning phases that are distinct from performance phases.

Developmental theorists from a wide variety of persuasions seem to agree on most of the following characterizations of cognitive development:

- performance in many domains appears to go through qualitatively different stages.

- some transitions are quite abrupt.

- relations between adjacent stages can be characterized in terms of hierarchical structures: the top-level units at one stage become the elementary units at the next higher stage.

- performance on a variety of tasks is rule-governed, even though early forms of rules in a domain are likely to be inadequate.

- childrens encodings of the environment tend to underrepresent some of its essential features. Development produces more complex representations.

- children tend to focus on local aspects of problem situations. Development produces a more global orientation.

- <add your favorite one here>

What appealed to me most about production systems was that they promised to provide a way to construct a theory of cognitive development that could account for these generalizations in terms of a completely unambiguously specified set of underlying information processes. additionally, the models would simultaneously be both performance theories and developmental theories.

By adopting a production system architecture, one could replace vague musing about the properties of cognitive development with well-targeted questions about the developmental course of specific pieces of the system architecture. In the good old days of PSG, the grandaddy of all running production systems, the questions were simpler than they are now with the proliferation of alternative architectures. But whether one adopts PRISM, or OPS5, or ACT*, one still has the advantage of being able to ask about the developmental course of: conflict resolution, wm protocols, variable bindings, complexity and depth of the match process, etc. At present there is a vast space of particular ways one can tune up a production system, so the developmental questions may have to await the outcome of extensive search in this space before they can be completely formulated. Nevertheless, the use of production systems to formulate developmental theory has what seems to me to be one of the hallmarks of scientific progress: a very hard question (what is the developmental mechanism?) has been replaced by several not-so-hard ones: (e.g., does the variable binding process change?, does the conflict resolution process change, etc.). The challenge to PDP models is to demonstrate that they provide an alternative way of simplifying and decomposing this fundamental issue.

Finally, I have two responses to Kieth's recent post.

On parallelism. Although the PDP folks have captured the sexy phrase "massively parallel", it is clear that PS theory (as distinct from PS implementations) are as massively parallel as there are productions in the system. All pds are supposed to be tested on every cycle. Indeed, one way to resolve this debate is simply to give the match process to the PDP folks, and ask them to deliver the variable bindings to the PS folks on the action side. No bindings, no cigar.

On necessity. In what way has the "necessity" of any of the K&T heuristics been demonstrated? It seems to me that one serious inadequacy of that line of work has been a failure to specify the conditions under which competing heuristics would be utilized. When does representativness dominate availability, etc.? When do none of the biases operate: eg. in many cases our judgements ARE veridical. The K&T work seems to suffer from the lack of just the sort of model that "some people" (KH?) would consider an academic exercise: that is, a model of underlying processes.

1-APR-85 13:48:14 shrager Goals and Readings
ource: shrager (jeff shrager @ cmu-psy-a)

I began to quote at length from Imre Lakatos' "The Methodology of Scientific Research Porgrammes" but decided to make the whole first chapter available in the lounge. There are two copies on the readings file (which will go into a folder as soon as I convince Walter to start one for the seminar).

I *highly* recommend reading this summary of various doctrines in the philosophy of science, and analysis of the way that "real" theories in "real" scientific research programs progress. A number of our most cherished scientific preconceptions are shown to be completely without philisophical support, and more importantly: without historical or current instantiations.

If you read no other part of this paper, please read the few pages in section 3 (pgs. 47-52) on Lakatos's heuristics of scientific research.

1-APR-85 21:24:53 mcclelland Some Reactions
ource: mcclelland (jay mcclelland @ cmu-psy-a)

Well, I can't resist contributing a wee bit to the ps/pdp discussion.

a) Several questions have been raised about the ability of PDP models to deal with the effects of things like explicit vs implicit learning, conscious indiction of rules from exemplars, and the effects of instructions on the tendency to do such consicious rule induction. Here, I would ("Cheerfully"?) say that PDP models are really much more applicable to the implicit learning situation than they are to the behaviour of sophisticated adult subjects consciously attempting to formulate rules to account for some set of exemplars in a psychology experiment on categorization. I want to be the first to acknowledge that people do often consciously formulate explicit rules and test them against instances to see if they hold. I happen not to think that much of this happens in things like language acquisition, so it seems to me that the PDP approach is quite applicable to that domain.

PDP models of learning are models of implicit learning. As far I I know no one has yet applied the PDP approach to the development of a model of explicit (when I say explicit here I mean conscious and reportable) hypothesize-and-test rule formulation. I actually believe that PDP models will have a lot to say about where hypotheses come from, but for now that would be a "just so" story.

b) Can PDP models learn at an abstract level? Absolutely! Are they embarassed by the fact that linear separability is generally not a pre-condition for learning? Not at all. I have to take some responsibility for contributing to the confusion here because of the fact that a particular pdp model that I introduced many of you to seemed not to be capable of either abstract learning or of learning (for example) to categorize when the categories were not linearly separable. I created a false impression -- these are not serious problems for the PDP approach.

There are several pdp models which learn to solve problems that are not linearly separable ones and build up abstract representations while doing so. I will not claim that these models have reached the stage of sufficiency to induce any needed structure to fit any given problem, but continuing progress is being made. How much initial architecture will be required, and how much training on components before dealing with larger wholes will be required remains to be seen in particular cases. I will be prepared to expand on these points should anyone be interested.

c) I want to make perfectly clear that I think that Production Systems are very valuable and should not be done away with completely, at least not until we've made a bit more progress with PDP. Seriously, PS models provide a much more compact notation than PDP models, and there are times when the PS approximation is completely adequate, just as there are times when it is profitable to discuss the properties of atoms without referring to their subatomic structure. Actually, the PS framework is sufficiently broad as to include all known PDP models as a subset. In general, though, people do not appear to be as powerful as they ought to be according to some production system formulations. Perhaps the constraints imposed by PDP are in fact reasonably accurate reflections of the limitations of human capabilities? This is obviously a very broad issue, and it will take much time to resolve, but it is worth pondering.

d) John listed three fundamental properties that he felt a mechanism ought to have in order to provide a sufficient framework for simulating cognition. I would like to dispell any possible implication that might have been derived from his note that such things are beyond the scope of PDP models.

> i) The framework must have a notion of goals as abstract objects controlling cognition.

In PDP models, goals are represented as node activations, or as patterns of activations over a set of units. These can have a very important controlling influence on the behavior of a network -- as much of an influence as, say, perceptual input. Examples of PDP models that clearly have goals are Gary Dell's paper in the recent special issue of Cognitive Science on Connectionism, as well as Rumelhart and Norman's typing model. These are real models, not just-so stories!

> ii) The framework must have explicit variables in it, to give generality to the rules of cognition.

Anderson suggests that there must be variables in the framework, but I think he would agree that the important point is to capture the observed generality of behavior -- whether explicit "variables" are necessary to do this is surely an empirical question.

The issue then is, do PDP models give sufficient generality, even without explicit rules? Indeed they do. A number of PDP models, including for example the verb learning model that Rumelhart and I have proposed, are explicitly designed to show how generalization to novel stimuli is the rule in pdp models. Though the model does not have what are conventionally called variables in productions, it effectively implements the following variablized productions:

```
IF x is the root form of a verb
and x consists of any sequence of phonemes
ending in a VOICED phoneme other than /d/
THEN the past of x is x+/d/.

IF x is the root form of a verb
and x consists of any sequence of phonemes
ending in a VOICED phoneme other than /t/
THEN the past of x is x+/t/.

IF x is the root form of a verb
and x consists of any sequence of phonemes
ending in t or d
THEN the past of x is x+/^d/.
```

Actually, those of you who are familiar with this model know that its behavior only approximates that which would be produced by the productions described above. In particular, it generalizes less well to novel words or pseudowords than to familiar words early in learning, even though it does show some tendency to generalize at this stage. While this would seem to be a deficiency from the point of view of rule-obediance, it seems to be a deficiency exhibited by actual kids learning the past tense. As we look at these things more closely we see that though people do exhibit some generality to the rules governing their behavior, this generality is perhaps

not as great as has often been suggested -- PDP models account for the hole as well as the donut!

> iii) It must be possible for the model to learn at the abstract rule level, rather than at the specific connection level.

This last objection is puzzling in view of the above. The verb learning model behaves (approximately) AS IF it were learning at the abstract rule level, even though in fact it is learning at the specific connection level.

My impression is that many Cognitive Scientists (and here I would list the likes of Fodor, Pylyshyn, vanLehn and many others) have failed to pay due attention to the difference between a description of the behavior of a cognitive system and the actual mechanism that implements the cognitive system. The problem lies, I think, with the lack of a formalism, until PDP models came along, which allows one to keep this distinction clear and be explicit about one's models at the same time. I am not sure whether this problem plays a role in John's comments, but I would like to address it anyway because it is a KEY POINT, and one of the issues that PDP models are attempting to make ALL OF US FACE. There has been a strong tendency in linguistics and some cognitive science circles to slide down the following slippery slope. Step 1 is the observation that some behavior is characterizable by a rule. Step 2 is to say that the rule is "in some sense" represented in the "competence" of the person whose behavior conforms to that rule. Step 3 is where the pitfall lies. Til PDP, there was simply no other way to be explicit about what one meant at Step 2 than to assert that the rule was really written down in the head somewhere. So, that's what people who wanted to be explicit did. In step 4, these same people propose mechanisms for acquiring these rules in this explicit form. Finally, in Step 5, they have forgotten altogether that explicitly writing the rules into the cognitive system might be only one of many different ways in which rule guided behavior can be modeled. PDP models are supplying an alternative here, by proposing that rule-characterizable behavior and the ability to learn to engage in such behavior CAN be explicitly modeled WITHOUT assuming that the rule is written down in the head somewhere. Having written the above I guess I have come around to the following response to John's point 3:

What makes PDP models interesting is precisely the fact that they CAN account for behavior that has previously required the explicit formulation of rules. No one would think that Kepler's laws are explicitly written down somewhere in the planets that obey them. Why then should we assume that the rules of language or any other cognitive domain are written down in our heads?

Jay McClelland

2-APR-85 00:37:01 schneider tenative agenda for meeting
ource: schneider (walter schneider @ cmu-psy-a)

DISCUSSION OF THE RELATIONSHIP OF PRODUCTION SYSTEM AND PARALLEL DISTRIBUTED PROCESSING FRAMEWORKS

Friday, April 12 in room 336B from 3:30 to 5:30 p.m.
participants: Anderson, Holyoak, Just, McClelland, Newell, Reder, Schneider
Organizers: Holyoak & Schneider
Tentative agenda:{} indicate some thoughts to get discussion going

0. Introductory remarks (5 min). Goal is to exchange viewpoints
1. Why is a part of the cognitive science community uncomfortable with the production system view of cognition? The goal is to get issues out in the open rather than defend or argue them(15 min)
   {descriptive language rather than theory; lack of constraint; difficulty of communication; complexity; limited microstructure tests; perceived lack of neural base; have not lived up to some expectations for learning; more brittle than the modularity claim}
2. What fundamental characteristics are assumed and what constraints are imposed when modeling from the Production System(PS) and the Parallel Distributed Processing(PDP) frameworks? How can they be falsified? (20 min)
   {PS are representations built from abstract objects, utilize variables, learn abstract rules rather than specific connections among elements, have limited working memory, and specific rules for conflict resolution
   PDP systems include connections between sets of neuron-like elements, all learning involves only changes in strength or thresholds of units, learning is done at a population level without hand crafting the weights, each unit is involved in many patterns, changes in a unit can have many indirect effects through the mass action influence}
3. Is the difference between these two views of cognition only a question of level of description? How much concern should the people at one level have about the problems at the other levels? {20 min}
   {PDP systems assume objects are points in a similiarity space operated on by many simultaneous constraints with associations between points in different spaces, rules come about through mass action as a result of learning experience
   PS are abstract symbol based condition action rules which can match lists of symbols in working memory and place new symbols in working memory
   an example of this not being a difference of levels: if the cognitive space has a metric distance and is not a set of abstract symbols then the PS system may have to invent a great deal of processing to perform operations available by the space metric}
4. What are the critical phenomena that are either the core or problem for each approach? What is the forte of each approach? (20 min)
   {PS variable binding, goal structured behavior, strategic behavior, language processing, learning by instruction and imitation,weak problem solving methods(e.g. what is the pattern in "94733749")
   PDP simultaneous mutual constraint, graceful degradation, speed accuracy tradeoff, implicit class generalization, behavior in accord with rules without specific rules, ability to output partial information (e.g. tip-of-the-tongue-phenomenon), rapid identification of low likelihood information (eg the fast no)}
5. What other approaches are there? How might we combine approaches? (20 min)
   {frames/scripts/schemas; hybrid: production system control of PDP units, production system operations over semantic nets; parallel

production systems with partial activation}

2-APR-85 11:10:02 anderson answers to the five questi...
ubject: answers to the five questions
ource: anderson (john anderson @ cmu-psy-a)

Given that Wally tells me that the meeting may be too free form for me to actually give answers to all the questions he fashioned, I thought it would be useful if I posted them since I did bother to compose them. The numbering corresponds to his questions though you will note I cannot help wandering a little afield at points:

1. I still object to this question although I now believe the authors did not intend the implication I object to. That implication is that PS are on the demise. I think production systems are an extremely complex and profound theoretical formalism. I don't think that when we first advanced them we understood all of their properties and potentials. For the majority of the history of production systems they have simply been ignored everywhere except at CMU. What we are seeing is a sign of success. Production systems can no longer be ignored. However, they are hard to understand and are often poorly communicated. I submit that every substantial objection to production systems are simply based on misunderstanding.

2. As I have argued both in post and in my book PS as a class are not subject to empirical falsification although particular exemplars like ACT* are most certainly. My major criticism of PDP systems is not that as a framework it is incapable of falsification, but rather that there does not seem to exist falsifiable theories within that framework. Consider my question about linear separability--I thought that maybe we had here a strength of PDP theories--maybe they did not predict linearly non-separable sets were impossible to learn, but at least that they were harder to learn. But Jay claims that PDP models make no predictions about linear separability.

I also think Jay misunderstood the three features I listed with PS models. He seemed to read them as three things I thought true of human cognition which PDP models could not account for. Far from it, I intended them as three non-trivial predictions of a certain class of PS theories (though by no means all) which might prove the key to falsifying them. These are of course rather abstract claims and so not subject to simple empirical test but perhaps subject to some complex reasoning from data. However, Jay concedes these features are true of human cognition and procedes to argue that PDP models can do it to.

3. On the third issue I can only repeat my first post: There are a number of analogies floating around for how to think of the relationship between the PDP level and the PS level. However, the one I would like to advance as not prejudging the issue is between machine language and a high-level programming language from LISP. At some level nobody doubts whether there is this machine-level PDP implementation, the empirical question is whether it is compiled from some higher level PS implementation. If it isn't we would expect there to be mental programs that do not perfectly correspond to any set of productions just as it is the case that not all possible machine language programs could be compiled from LISP programs. On the other hand if there is a reality to the PS level we should discover that there exists no PDP level system that does not implement a production system. Thus, at this abstract level the issue seems quite decidable.

There are a number of qualifications on this of course. One is that the PS level has a limited range of applicability. Nobody proposes to model what is happening on the retina in terms of PS systems for instance. Second, the

reality of the PS level does not deny that there are phenomena that require digging below the production. To understand the exact timing properties of cognition or exactly how it breaks down one will have to dig below the production level even as one needs to dig below LISP code to understand these issues in a computer. However, this fact does not deny the truth of the claim that there is a level of understanding at which the LISP code is a complete specification of the system and one does not need to look below. Looking below only complicates thr story.

4. Question 4 asks whether there are empirical phenomena that are particularly suited to one approach. I will answer a slightly different question: whether there are empirical phenomena that are key predictions of one approach. PS systems like ACT make certain "abstract" predictions which might be worth putting to test. These were the three features that Jay conceded. I don't think he should have.

a) Goals: Goals such as in the typing model only have a family resemblance to the use of goals in the ACT* theory and other PS theories where there was an explicit flow of control, popping mechanism, etc. Of the three assumptions I think this is the most problematic in ACT*. So I would advise against trying to conjure up a PDP model to reproduce these properties.

b) Variables: Just to say a system has variables of course is to say nothing about its computational properties. Variables derive much of their computational power and empirical punch when they are combined with relational data structures which the verb example does not have. This is the kind of production that it would be interesting to see implemented in PDP models:

```
If the goal is to add digits n1 and n2 in column1
   and n1 + n2 = n3
   and n3 > 9
   and 10 + n4 = n3
   and column2 is to the left of column1
THEN write n4 in column1
     and carry 1 to column2
     and make the new goal to add the digits in column2

Where n1,n2,n3,n4,column1, column2, are the variables
```

Everyone's firm believe is that all of language is not characterized mentally as finite state grammar but the example of past tense inflection is a finite state phenomenon. I don't think we can capture the non finite state phenomenon without relational rules plus variables in a PS system.

c) Abstract Rules: Actually Jay does not concede abstract rule learning. He notes that it is a non trivial step to go to the assumption that there are such abstract rules in the head. This is precisely the point! It is a trully profound prediction of the PS framework that these rules are not descriptive fictions but are real. If it is true it would rank as one of the greatest scientific discoveries of all time and tell us more about our nature than anything else we currently know. While a specialization of Newell's claim about physical symbol systems it shows us just how profound a claim Newell is making. Note also that in PS systems unlike linguistic theory these rules are part of an explanatory process theory. The claim is that just as LISP programs evolve and run according to the abstract rules of LISP so the human' mind develops and runs according to the abstract rules of productions systems. Hallelujah!

The problem is that it is hard to distinguish between a mind behaving according to abstract rules and one that just is described by abstract rules. However, this is just where the learning theory takes hold--while it is hard to distinguish beteen finished products, systems that learn in units of abstract rules look very different than systems that learn according to PDP principles. This is fundamentally an empirical question but everywhere I have looked with care it comes out in favor of the abstract rule point of view. You would never be able to account for LISP learning by simple PDP mechanisms. I also suspect you would never be able to account for second language learning with PDP mechanisms and there are a lot of empirical communalities between second and first language learnings including the errors that feed the verb model.

5. I think all learning models fail to represent correctly the role of analogy in learning--particularly, the highly strategic and problem-solving character of analogy use. I also think that existing systems including ACT fail to represent the flexibility of flow of control and how that flow of control is anchored in the structure of the physical problem. I think VanLehn's proposal for annotated grammars comes closest to a solution on this second issue. For the right solution to both problems, stay tuned for the PUPS release.

2-APR-85 14:35:33 mcclelland views on the 5 questions
ource: mcclelland (jay mcclelland @ cmu-psy-a)

Answers to the 5 questions raised by schneider/holyoak from one PDP person:

1) Why are some people dissatisfied with production sysytems?

Speaking for myself, I found that I was able to get more directly at the heart of the problems I wished to simulate without taking on the overhead associated with any particular production system formulation that already existed or at least they I knew about and had access to. As I said before, though, the ps framework includes pdp models as a subset so this is a comment about particular existing ps models rather than a problem with the framework itself.

2) What are the characteristic assumptions and constraints on each approach?

I can comment on pdp models as follows: A pdp model is a model consisting of a large number of computational elements.

Elements have:

an activation at time t
an output at time t
connections to other units.

connections are:

places where the outputs of 1 or more units are multiplied together with each other and with a connection weight to provide one of many inputs to another unit.

Processing is:

the evolution of the pattern of activation over the set of units in the model over time. This evolution occurs according to a set of equations which specify hoow the sum of the weighted inputs to each unit influences its activation. Activations may be updated synchronously or asynchronously; if the latter, some assumption is required about order of update.

Learning is:

the change in strengths of the connections in the model over time. These changes are made on the basis of information locally available to the connection; however in practice this isn't much of a restriction since it is possible to construct a newtork in which some global quantity is comptuted by some overseer unit and then is sent via some special input to each unit in the model. [I am not happy with this but the capability is there for doing these kinds of things].

3) Is the distinction between PDP and PS only one of level?

No. One reason is that PS can be used to characterize all levels, while PDP can only be used to characterize the microstructure

level. This is as consequence of the fact that PDP is a subset of PS.

Here is a proposition:

I. PDP is a sufficient framework for characterizing the Microstructure of cognition.

What I mean by this is that we need not introduce the concepts of variables, abstract rules etc directly into the microstrutural level of description -- the more restrictive framework above is sufficient at this level. Note that I am not saying that these concepts are not valuable at a higher level. I'm saying that they are not required at the microstructural level. This is analogous to saying that the notion of a double bond need not be a part of one's theory of the internal structure of the atom.

Here, though, is an important question: To what extent should we believe that PDP is anything other than an implementation of some higher level language?

According to the PDP approach, there is no reason whatsoever to suppose that what is learned in the connection strengths will actually be the compiled version of some higher level program.

Actually, I think that in some cases we should believe that PDP is the implementation of something that is "compiled from" a higher level language and in some cases we should not.

I hereby offer a second proposition:

II. It is appropriate to believe that PDP is just an implementation of something compiled from a higher level language if and only if one is speaking of knowledge represented in the form of a consciously reportable verbal proposition.

A related empirical question is:

> Under what conditions is it correct to assume that the knowledge a person has in a particular domain is encoded in the form of consciously reportable verbal propositions?

If we assume that (many aspects of) expert knowledge is proceduralized and is not in explicit verbal form (though there may or not be explicit verbal propositions in the expert's head) then we can ask the following question:

> Under what conditions can proceduralized expert knowledge be aquired without having first learned (either by self-instruction or external tutelege) a set of consciously reportable verbal propositions?

Discussion:

We can learn explicit rules, and of course they affect our behavior -- when we learn them in such a way as to be able to recall them in the appropriate

circumstances. A pdp model could be used to implement the actual learning of the rule and the association of that rule with appropriate contextual cues so as to allow it to be recalled when appropriate. Surely this is what goes on when a bridge player learns "When the board is on the right, lead the weakest thing in sight". As someone who has played bridge with too many novices, I can tell you that they nearly always fail to remember this rule when the time comes. But, if they did, they would be able to use it to guide their selection of what suit to lead next. Clearly in this case the pdp model is very much the implementation of the learning of the explicit verbal proposition that is the rule. PDP might provide some insight into such things as how easily the rule could be recalled and under what circumstances and how easily it would be confused or blended with other rules.

On the other hand the actual procedural learning underlying behavior in accordance with the rule is quite different from the knowledge of the statement of the rule. When a good bridge player plays bridge he does not think of this rule explicitly. This sort of "compiled knowledge" is the basis of real skilled behavior.

OK McClelland and Anderson agree here -- but wait, there is still a crucial issue. As I understand it Anderson assumes that compiled knowledge depends upon having previously had the knowledge in an uncompiled state. McClelland disagrees with the necessity of this assumption. I would make the following points: a) I don't think first language learning really involves first explicitly formulating the rules and then compiling them into the production apparatus. b) several studies suggest that attempts to explicitly formulate rules often get in the way of acquiring a "feel" for an (artificial) language. Could this have anything to do with the fact that I always got straight A's in all my foreign language classes but could never so much as order a meal in a German restaurant?
c) But this doesn't mean that it won't help a great deal to have an explicit verbal rule to tell you what to do in certain cases. Almost certainly Lisp learning is one of those cases.

I would like to comment on propositions I and II: Surely they need refinement. I hope as these discussions continue (over the years -- I don't expect a resolution this afternoon) that more accurate propositions can be articulated.

4) What are the core phenomena of each approach and what are the key problems? Here I'll just list core desireable properties of PDP models. Of course a PS can be written with all of these properties:

a) PDP models provide a natural framework for handling large numbers of simultaneous, mutual constraints among different pieces of information. In this regard they seem extremely well suited for perceptual processing, for selection of the best action to perform in a complex situation, and for other constraint satisfaction problems.

b) PDP models degrade gracefully with damage and with noise in their inputs.

c) PDP models can learn through doing without already being smart -- they do not require the ability to formulate explicit rules to come to act in accord with such rules.

What are the key problems with the approach?

a) Though slightly more constrained than PS in general, PDP has considerable expressive power and is itself too unconstrained. I agree completely that both PS and PDP are simply frameworks for models and not models per se. Particular PDP models are of course just as falsifiable as Particular PS models. Indeed, I've formulated many which clearly held no water at all -- surely there is no serious doubt that it is possible to construct particular falsifiable models within the PDP framework. In fact, I think it's probably true that even the best models I know of to date are clearly false in some detail. Unfortunately it is not possible in general to falsify particular individual assumptions of particular PDP models, one can only say that a particular entire constellation of assumptions is false. However, this is no less true for PDP than it is for anyone else.

b) Since the approach is concerned with the microstructure of cognition, there is a lot that it leaves out. Explicit assumptions have to be made about the macrostructure. Unfortunately the macrostructure of the mind is the product of a very large number of different constraints acting over evolutionary time. Certain general principles may hold true, and may even be derivable in some cases from characteristics of the microstructure, but this may not necessarily be the case.

5) What other approaches are there? Can/Should PS and PDP be combined?

I'll just say something about the second part of the question.

I believe that PS models make a pretty good high-level language for giving exact descriptions of abstract rules where these really are implemented by the microstructure and for giving approximate descriptions of what's really going on when such rules are not very exact characterizations. This means that I am entirely open to hybrid models. For example, implementing external control over a PDP network in a production system is perfectly reasonable. An issue left open in so doing would be simply this: Would a PDP model of the external control provide a more exact description of the external control, or would it only implement the external control expressable exactly in the productions? Obviously, this would depend on the particular production system used.