Particular cases provide us guidance in both the formulation and comprehension of general laws.

Lawler's Focus: Learning through Interaction

It is remarkable in Feynman's discussion that something so familiar as reflection from a mirror is a pathway into solving the deepest puzzles of physics, as in wave-particle duality.[18] And yet, why be surprised ? Insights are usually a reconceptualization of familiar affairs. Learning is something we have all experienced personally for long periods of time, something we see in others all the time. There are epistemological and psychological reasons to believe that a case-based approach is better suited than lab-based methods for gathering information about developmental issues, such as the character of learning.[19] Specifically, if one sees learning as an adaptive developmental mechanism, then one should look at learning where it happens in the everyday world. Furthermore, if learning is a process of changing one state of a cognitive system to another, then representations of that process in computing terms should be expected to be more apt than in other schemes

---

examined, one has to specify only the properties of the objects which are to be used. These properties are placed as axioms at the start. It is no longer necessary to explain what the objects that should be studied really are."

N. Bourbaki, in J. Fang, p. 69.

[18] See the later discussion under the heading "Feynman: on the Reality of Reflection All Ways."

[19] If one recalls that case study is the method underlying the theories of Freud and Piaget and that ecological studies such as those of Barker and Wright (1951) and, in our our decades, the stunning work of Goodall (1971, 1990), it is not hard to believe that the fusion of such methods may continue to help us learn about human development. Ecologically oriented studies, such as those of Barker and Wright (1951, 1967), pay close attention to the context of behavior. That context of behavior is also the primary situation in which learning takes place and thus should be considered in detail in any study of learning through interaction.

where the procedural element of representation might be less important.[20] One might take details of case study corpora much as one uses the boundary conditions to specify the particular form of a solution to a differential equation. Such is the use made of the psychological studies here, as a foundation for the representations used in the computational models, and as justification for focussing on key issues: the role of egocentricity, and the inception of multi-role play by one agent as a key event.

I ask how learning is possible at all, especially if one takes seriously the notion that knowledge depends on the particular details of experience; and even more, how one can learn from interacting with an agent one certainly doesn't understand, and probably doesn't even pay attention to ? [21] We know such learning happens, for we see it in people every day. How could such learning happen in a machine ? [22] The acquisition of knowledge can be explored through machine-based modelling at a level of detail not previously possible either through introspection or through observation.

Must we claim that learning happens in people the same way? Not necessarily, but the virtue of machine learning studies is that such epistemological mini-theories allow us no miracles; they can completely and

---

[20] A straigtforward assertion of this point can be found in Minsky's Turing Award Paper, Form and Content in Computer science, (1970). Minsky, in a talk to graduate students in his lab (1976), mentioned that a major problem AI needed to solve was how the control structure of mind emerged from the processes of its functioning.

[21] This involves the notion of ego-centrism in young children (Piaget, 1926) and the parallel problem that one shouldn't expect an agent to comprehend adequately another more sophisticated than itself.

[22] Putting knowledge in machines forces us to confront the problem directly. You can look inside a machine and tell precisely where the knowledge comes from. Perhaps you will decide "the programmer put it here"; or perhaps "the programmer built a rule system from which the new knowledge could be inferred". You may even ultimately decide "the programmer exploited the characteristics of the machine or the implications of the representational scheme to give an appearance of novelty there,". (Such is the argument in a paper presented at the AAAI conference in 1983 by Brown and Lenat about Lenat's well known system AM, Lenat, 1979.)

unambiguously cover some examples of learning with mechanisms simple enough to be comprehensible. Building epistemological models is an exploration of the possible, according to a specification of what dimensions of consideration might be important. The computer's aid in systematically generating sets of all possible conditions helps liberate our view of what possible experiences might serve as paths of learning. Feynman asked what happened at all those other places on the reflecting surface where the angle of incidence doesn't equal the angle of reflection. Similarly, when we can generate all possible interactions through which learning might occur–including some we imagine are not important, not only can we explore those alternate paths, we must do so, especially if they don't follow the party line of our own best current theory.

Considering All the Possibilities

The analysis that I choose to contrast with Feynman's arose as a verbal theory [23] of one child's learning strategic play at tic-tac-toe. It was continued in a constructive mode through developing a computer-embodied model, SLIM (Strategy Learner, Interactive Model). The latter was based on search through the space of possible interactions between one programmed agent having some of the characteristics of the subject, my daughter, and a second, REO, a programmed **R**easonably **E**xpert **O**pponent. REO is *expert* in the sense of being able to apply uniformly a simple set of rules for good tactical play. [24]

---

[23] This analysis was published as chapter 4 in Lawler (1985) and as a memo of the MIT Artificial Intelligence lab, Lawler (1980). It is a detailed analysis of the complete history of one child's play at tic-tac-toe throughout a two year period. The subject of the study was my daughter Miriam.

[24] The set of rule REO follows for each execution are, in order, these:

1. if it is possible to win --> make the winning move.
2. if the opponent can win --> block at least one threat.
3. if the center cell is blank --> move in the center cell.

Strategies for achieving specific forks are the knowledge structures of SLIM. I represent each as having three parts: a **G**oal, a sequence of **A**ctions, and a set of **C**onstraints on those actions (each triple is thereby a *GAC*). I simulated such structures in a program that plays tic-tac-toe against variations of REO.[25] Applying these strategies leads to moves that often result in winning or losing; this in turn leads to the creation of new structures, by modifying the current GACs. The modifications are controlled by a small set of specific rules, so that the GACs are related by the ways modifications can map from one to another. Subject to certain limitations, I've completely explored certain classes of strategies.[26]

The study avoided abstraction, in order to explore learning based on the modification of fully explicit strategies learned through particular experiences. The results are first, a catalog of specific experiences through which learning occurs within this system and second, a description of networks of descent of concrete strategies from one another. The catalog permits a specification of 1). which new forks may be learned when some predecessor is known and 2). which specific interaction gives rise to each fork

---

4. if a corner cell is empty --> move in a corner cell.

5. if a side cell is empty --> move in a side cell.

For a slightly more sophisticated example of a production system which would play following a similar set of rules, see Human Problem Solving, Newell and Simon, (1972), p.62, Figure 3.4

25 The "variations" of REO are created by crippling specific rules representing REO's ordered preferences for moves to the center cell (first preference), corner cells (second) and the remainder. When rules 3 and 4 above are crippled, REO will choose any free cell; the preferences are <u>unstructured</u>. When only rule 3 is crippled, REO will select first a corner cell or the center cell and, when none of those are free, select a side cell as a default choice. In this case REO's preferences are <u>structured</u>. When no rules are crippled, REO always prefers an empty center cell to others and prefers an empty corner cell to side cells. In this case, the preferences are <u>highly structured</u>.

26 The strategies explored focus on games in which SLIM plays the first move in cell 1. This specific preference is based on occurrences in the case study and also on the fact that such corner-opening play is the richest micro-domain within tic-tac-toe play.

GACs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
games	2	2	1	1	1	1	6	6	2	2	6	6	1	1
wins	2	2	0	0	0	0	2	2	0	0	2	2	0	0
losses	0	0	1	1	1	1	0	0	0	0	0	0	0	0
draws	0	0	0	0	0	0	4	4	2	2	4	4	1	1
new GACs	**1**	**1**	0	0	0	0	**2**	**2**	0	0	0	0	0	0
constraints	0	0	1	1	1	1	0	0	0	0	0	0	0	0

This is a table of wins and losses for each of 14 GACs

learned. The result obviously also depends on the specific learning algorithm used by SLIM.

The Path of Learning

Table I shows how limited is the learning possible for SLIM playing against REO when the latter is fully committed to playing with the complete set of rules (REO has highly structured preferences). Forking patterns are represented as lists of cells in the 3x3 tic-tac-toe grid, numbered 1 to 9 from left to right and top to bottom. SLIM's strategies are represented as a list of triples. First is a group of three cell numbers representing a pattern for a fork. This is the Goal, e.g {1 3 9}. Second is an ordered list of those three numbers, showing the sequence in which moves are to be made. This is the Action-plan, e.g. [1 9 3]. Third is a list of constraints -- used to inhibit further use of the plan when it is found faulty. The triple is the **GAC** (for **G**oal, **A**ction-plan, **C**onstraint), e.g. GAC 1 is represented as **[** {1 3 9} [1 9 3] [nil] **]** when play begins and no constraints exist. SLIM starts every simulation knowing a single GAC. Notice that beginning with either GAC 1 or 2 only a single new forking strategy is learned. Two others are learnable with either GAC 7 or 8. No other forking strategies are learned.

SLIM's Learning from Interaction: An Example

In order to evaluate specific learning mechanisms in particular cases, one must go beyond counting outcomes; one must examine and specify which forks are learned from which predecessors in which sequence and under which conditions. Within the virtual universe of SLIM's simulation, consider how the symmetrical variation to one particular fork can be learned. Suppose that SLIM begins with the objective of developing a fork represented by the pattern {1 3 9} and will proceed with the plan [1 9 3].[27] SLIM moves first to cell 1. No problem. REO prefers cell five, the center cell, and moves there. SLIM moves in cell 9. Everything goes according to plan. But now, REO's second move is to cell 3. SLIM's plan is blocked. The strategic goal {1 3 9} is given over -- but the game is not over. SLIM, playing tactically now with the same set of rules as REO, moves into cell 7, the only remaining corner cell. Unknowingly, SLIM has created a fork symmetrical to its fork-goal. SLIM can not recognize the fork. It has not the knowledge to do so. What happens ? REO blocks one of SLIM's two ways-to-win, choosing cell 4. SLIM, playing tactically, recognizes that it can win and moves into cell 8. This is the key juncture.

A		2
3	1	
C	D	B

SLIM moves first. SLIM's moves are letters; REO's are numbers.

When SLIM wins a game without expecting to do so, it recognizes that the circumstance is special; even more, SLIM assumes that it has won through

---

[27] The implication here is that SLIM will move first to cell 1, next to cell 9, and achieve the fork-goal by moving to cell 3.

creating an unrecognized fork (otherwise REO would have blocked the win). What could that fork be ? SLIM takes the pattern of its first three moves as a fork. That pattern is made the goal of a new GAC. SLIM examines its known plans for creating a fork (there is one; moves [1 9 3]) with the list of its own moves, executed in sequence before the winning move was made [1 9 7]. The terminal step of the plan is the only difference between the two. SLIM modifies that plan terminal step to create a new plan, [1 9 7]. SLIM now has two GACs for future play. [28] We know the fork achieved by plans [1 9 3] and [1 9 7] are symmetrical. SLIM has no knowledge of symmetry and no way of knowing that the forks are related other than through descent, that is, the derivation of the second from the first. [29]

---

[28] This example shows modification of the terminal step of the plan. A second kind of learning removes the middle step of the plan; this is center deletion. When a new cell is added to the plan after center deletion, two new plans are created reflecting the two orderings of the last two steps.

[29] One could call this type of machine learning "reflective plan construction". It is part of a long line of development which began with machine learning based on composition of primitives or subprocedures in the sixties. I built my own first compositional models following Selfridge's notions on learning counting. In that paradigm, an instructor set a goal for a system to reach. For every primitive and procedure it knew, the system reset the condition of the world to its initial state and executed each in turn, then checked to see if the world state had changed so as to meet specification of the goal. If they all failed, it began to try their combinations systematically. When some combination of primitives and procedures changed the world state to satisfy the goal, the system created a new procedure of those primitives and procedures executed in the successful sequence. Satinoff (Science, 1978) presents a physiological example of the integration of disparate systems for thermal regulation as an example of emergent behavior. This, of course, represents an analagous, biological model of evolution through composition of predecessor systems. Piaget (Biology and Knowledge, 1967, p. 321) cites much earlier biological examples by Jackson and Sherrington. Lawler, 1979, 1985, assumes the disparateness of cognitive structure and interpretes learning in the human case through their interrelation. Such an explanation of learning can also be seen in Papert's notion that Piagetian stages could be explained by the insertion of controlling agents in networks (Mindstorms, 1980, chapter 7), a theme he discussed as early as the sixties according to Minsky (Society of Mind, 1986, p.102).

The Results of a Simulation with SLIM:

The primary result of SLIM's simulation is an extended listing of the the new forks learned and the specific games through which they were learned given SLIM's knowledge before the game. The game where GAC2 was learned through tactical play after blockage of GAC1 would appear in such as list this way:

Original GAC	Learned GAC	List of Moves in Game	Process
(1 3 9)[1 9 3][]	(1 7 9)[1 9 7][]	[1 5 9 3 7 4 8]	PTM

An extended list of new fork-goals generated in particular games is adequate for specifying detail but somewhat opaque. It certainly does not highlight the interrelationships among the various forks which represent their genesis from one another.

Re-Representing the Results:

One can represent these learning paths as a tree of descent showing which forks may be learned from given prototypes. Consequently, one can make a tree that specifies the descent of all forks learnable from each prototype fork. Figure 3 exhibits the learning outcomes tallied in Table I. The number triplets are the action-plans for forks. It shows also that against a rule-driven opponent, SLIM was able to learn very little. Specifically, starting with GAC 1, SLIM was able to learn only GAC 2 and vice versa. Only GAC 7 (and its symmetrical variant, GAC 8) precipitated the learning of other strategies. Consider Figure 3.

GAC 7, which is the goal most productive of new learned strategies, does not provide an expert win. The plan of GAC 7 merely sets traps for an unwary opponent. But playing with GAC 7's fork as its goal, SLIM was able to learn more than from using an expert win as a model. The less effective goal led SLIM to exercise more alternatives which, in turn, permitted SLIM to learn more new forks; these new forks included two that it could not learn against this opponent from knowing either of the more effective strategies of GACs 1 and 2. These epistemological conclusions help explain why learning through experience–with all its attendant errors and mistakes–is nonetheless so effective.

The complete set of results then involves consideration of all paths of possible learning, even those deemed unlikely a priori, and concludes with the complete specification of all possible paths of learning every fork given any fork prototype. Consideration of all paths of learning I take to be comparable to consideration of all paths over which photon might travel when striking a surface.

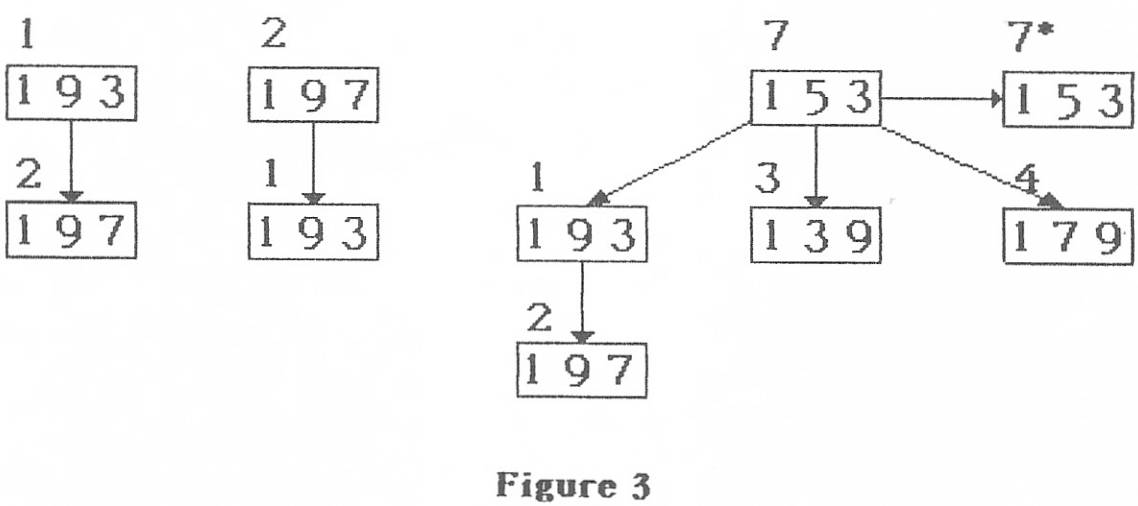

Figure 3

Descent tree for GACs 1 & 2   **This is a descent tree for GAC 7 vs. a high structure opponent**

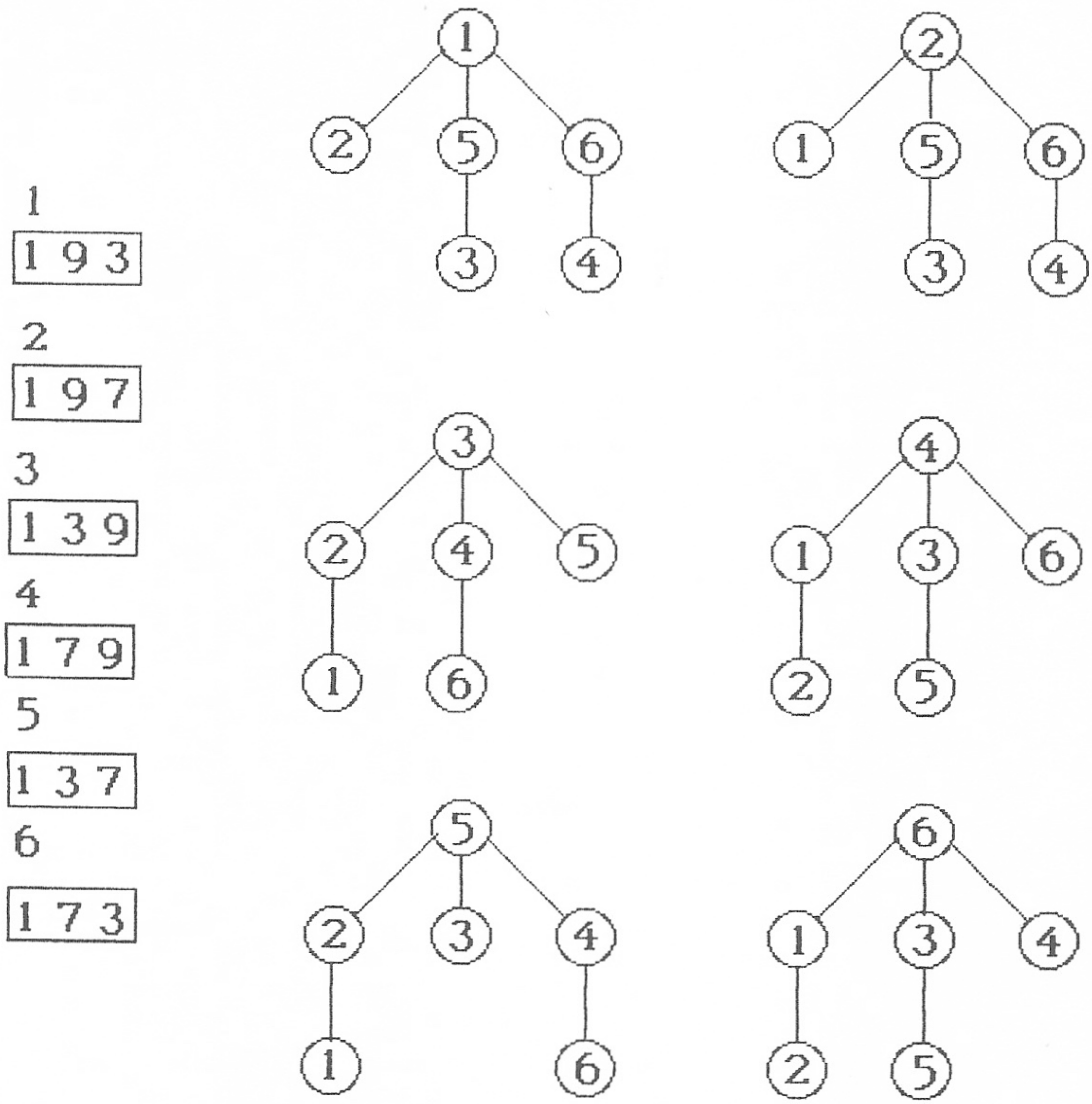

Figure 4
Trees of Strategy Descent from prototype forks 1-6 Based on Play Against a Structured Opponent

<u>Aggregating Results</u>

The first six GACs (shown in Figure 4) form a central collection of strategies. The tree with strategy three as top node may be taken as typical. Play in five specific games generates the other five central GACs. The specialness of the six central nodes is a consequence of their symmetry in respect of cogenerability. Some of those directly generatable can generate each other; they are reciprocally generatable. Those that are remotely generatable nonetheless lead to each other through intermediaries; they are cyclically generatable. For these six central strategies, the trees of structure descent can fold together into a connected network of descent whose relations of co-generativity are shown in Figure 5. [30]

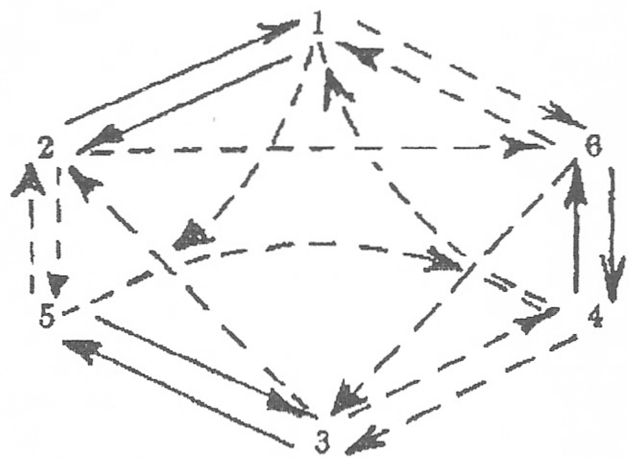

Figure 5 (redo ASAP)
Strategy Descent Network: Structured Opponent

Figure 6 summarizes the results of comparable simulations of play against an opponent with minimal cell-preference rules. In this mode of play, where knowledge-based preferences of REO are crippled, the "next move" decision is reduced to random choice unless either a win is immediately possible or an opponent victory must be blocked. In practice, for a program which generates all possible games, play against the unstructured opponent does generate all games, including some that would be most surprising if played by a knowledgeable opponent.

---

[30] Notice that those strategies cyclically generatable form the interconnection by GCD (Goal-guided Center Deletion) between themselves and others reciprocally generable by either PTM (Plan Terminal Modification) or GCD. In the figure, solid arrows show descent by PTM; dashed arrows show descent by GCD.

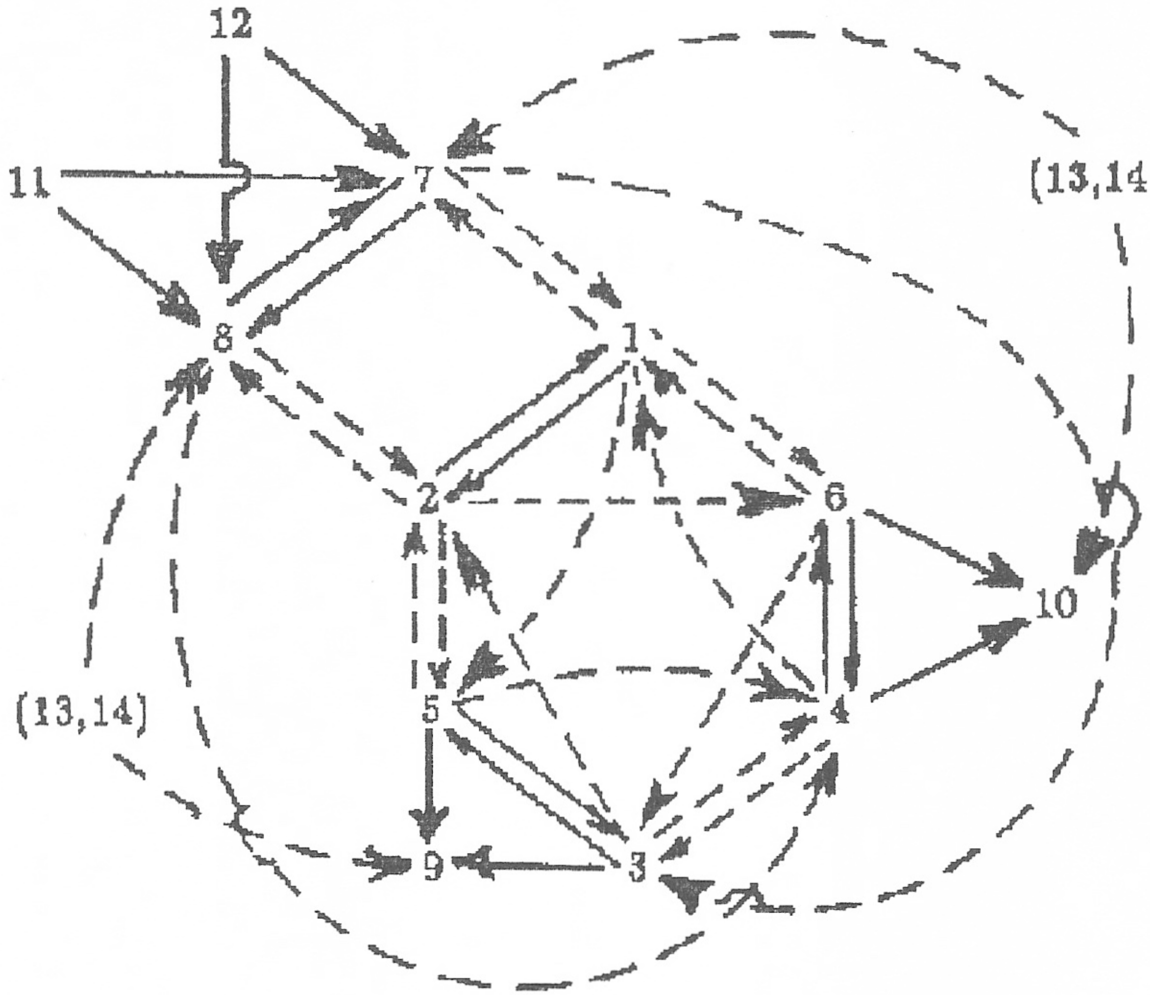

Figure 6

Strategy Descent Network:Unstructured Opponent

Extended Caption: GACs 1 to 6 still remain central, but the flexibility of the unstructured opponent permits SLIM to learn additional strategies. GACs 13 and 14 appear twice to simplify the network drawing. GACs 13 and 14 remain non-learnable through experience with this opponent. The specific reason is that SLIM's tactical preferences remained structured in these simulations. Thus SLIM, never trying games whose second move is to a side cell, will never win accidentally with a fork containing such patterns; consequently, SLIM can never learn such strategies.

The form of these descent networks is related to symmetry among forking patterns. But they include more: they reflect the play of the opponent, the order in which the forks are learned, and the specific learning mechanisms permitted in the simulations. These descent networks are summaries of results.

Comparison and Contrast

The apparent similarities relating Feynman's analysis of reflection and the exploratory epistemology of SLIM occur at different levels. They begin with a focus on detail:

- in the analysis of specific cases and
- in the analysis of the interaction of objects or agents with their context.

The core principle applied in both is to try all cases and construct an interpretation of them. There are many paths of possible learning, some central and some peripheral. In QED, the criterion of centrality is near-uniform directionality of the photon arrows. In SLIM, the criterion of centrality is a different and a new one: **co-generativity**. The core method is to aggregate results of all possibilities in a fully explicit manner. The process of aggregation is where the differences become systematic and significant. In QED, the aggregation of individual results is formally analytic–that is the solution of path-integral equations of functions of complex variables [31]. In

---

[31] The Preface to *Quantum Mechanics and Path Integrals* (1965) provides a little history which puts the descriptions used here in their proper perspective:

"The fundamental physical and mathematical concepts which underlie the path integral approach to quantum mechanics were first developed by R. P. Feynman in the course of his graduate studies at Princeton.... These early inquiries were involved with the problem of the infinite self-energy of the electron. In working on that problem, a "least-action" principal using half-advanced and half-retarded potentials was discovered. The principle could deal successfully with the infinity arising in the application of classical electrodynamics.

The problem then became one of applying this action principle to quantum mechanics in such a way that classical mechanics could arise naturally as a special case of quantum mechanics when h was allowed to go to zero.