inference in restricted versions of first-order logic (see Chapter 10). Formal probabilistic analysis of the expected payoff of EBL can be found in (Greiner, 1989; Subramanian and Feldman, 1990). An excellent survey appears in (Dietterich, 1990).

ANALOGICAL REASONING

Instead of using examples as foci for generalization, one can use them directly to solve new problems in a process known as **analogical reasoning**. This form of reasoning ranges from a form of plausible reasoning based on degree of similarity (Gentner, 1983), through a form of deductive inference based on determinations (Davies and Russell, 1987) but requiring the participation of the example, to a form of "lazy" EBL that tailors the direction of generalization of the old example to fit the needs of the new problem. This latter form of analogical reasoning is found most commonly in **case-based reasoning** (Kolodner, 1993) and **derivational analogy** (Veloso and Carbonell, 1993).

Relevance information in the form of functional dependencies was first developed in the database community, where it is used to structure large sets of attributes into manageable subsets. Functional dependencies were used for analogical reasoning by Carbonell and Collins (1973), and given a more logical flavor by Bobrow and Raphael (1974). Dependencies were independently rediscovered, and given a full logical analysis, by Davies and Russell (Davies, 1985; Davies and Russell, 1987) for the problem of analogical inference. They were used for declarative bias by Russell and Grosof (1987). The equivalence of determinations to a restricted-vocabulary hypothesis space was proved in (Russell, 1988). Learning algorithms for determinations, and the improved performance obtained by RBDTL, were first shown in the FOCUS algorithm in (Almuallim and Dietterich, 1991). Tadepalli (1993) describes an ingenious algorithm for learning with determinations that shows large improvements in learning speed.

The study of methods for learning first-order logical sentences began with the remarkable Ph.D. thesis by Gordon Plotkin (1971) at Edinburgh. Although Plotkin developed many of the theorems and methods that are in current use in ILP, he was discouraged by some undecidability results for certain subproblems in induction. MIS (Shapiro, 1981) reintroduced the problem of learning logic programs, but was mainly seen as a contribution to the theory of automated debugging. The field was reinvigorated by Muggleton and Buntine (1988), whose CIGOL program incorporated a slightly incomplete version of inverse resolution and was capable of generating new predicates.[4] More recent systems include GOLEM (Muggleton and Cao, 1990), ITOU (Rouveirol and Puget, 1989) and CLINT (De Raedt, 1992). A second thread of ILP research began with Quinlan's FOIL system, described in this chapter (Quinlan, 1990). A formal analysis of ILP methods appears in (Muggleton, 1991), and a large collection of papers in (Muggleton, 1992).

Early complexity results by Haussler (1989) suggested that learning first-order sentences was hopelessly complex. However, with better understanding of the importance of various kinds of syntactic restrictions on clauses, positive results have been obtained even for clauses with recursion (Dzeroski *et al.*, 1992). A recent paper by Kietz and Dzeroski (1994) provides an excellent survey of complexity results in ILP.

DISCOVERY SYSTEMS

Although ILP now seems to be the dominant approach to constructive induction, it has not been the only approach taken. So-called **discovery systems** aim to model the process of scientific discovery of new concepts, usually by a direct search in the space of concept definitions. Doug Lenat's AM (Automated Mathematician) (Davis and Lenat, 1982) used discovery heuristics

[4] The inverse resolution method also appears in (Russell, 1986), where a complete algorithm is mentioned in a footnote.