Acknowledgments

First and foremost, this thesis owes its shape and content to my advisor, Dana Angluin. She has directed my research since I came to Yale, and has had a great influence on what I have accomplished in these years, and what I know today. Her thoroughness and sincerity in doing this will be a model for me in the years to come. She also tried to teach me her high professional and personal standards; although I resisted as strongly as I could, there are still some traces of them in this thesis.

Drew McDermott escorted my first steps in Prolog. He gave me some insightful suggestions, which made me wish he had given me lots more. Just to name a few, the definition of incremental inductive inference, the idea of mechanizing oracle queries, and the methods that led to the algorithm that diagnoses finite failure, were all originated in some off-hand comments he made when he failed to avoid me in the corridor.

Alan Perlis and Mike Fischer also contributed to the final form of the thesis, especially to its introduction and conclusions. Without Alan's encouragement, I would not have had the nerve to make some of the bold claims I have made, and his flamboyant manner helped to counter-balance Dana's caution and serenity.

I got a lot of support from the people of the logic programming community. I have learned a lot from discussions with Lawrence Byrd, Ken Bowen, Alan Colmerauer, Maarten van Emden, Bob Kowalski, Frank McCabe, Fernando Pereira, and David Warren, to name a few.

David Warren's Prolog implementation was an indispensable research tool. First across the ocean, and then across the Arpanet, he reminded me that I am not the only Prolog hacker in the world, even though that's how it feels at Yale.

Donald Michie's enthusiasm about my research was an ultimate source of pleasure, and helped me feel that what I am doing may be worth while. Together with Alan Perlis, he showed me that intellectual curiosity and openness are ageless and statusless.

Bob Nix kept me informed on what is going on in the world, while I was too preoccupied with my work to pay attention to anything else. I think I have learnt more about computer science from discussions with him than from all of the courses I have taken together. I wish I could take him with me as an office mate wherever I go. He also devoted a lot of his time to reading and correcting my thesis, which compensated for some of my ignorance of English and Combinatorics.

Many people made specific technical contributions to the thesis, which I would like to acknowledge: Frank McCabe was the first to suggest that the query complexity of the diagnosis algorithms may be improved; only after this realization, did I begin to believe that they can be more than an exercise in formalizing some vague intuitions, but a real debugging tool. He also found how to diagnose Prolog programs with negation.

Ryszard Michalski insisted on understanding what I was doing, which made things clearer to me too; he offered some terminological improvements that made the formal machinery I used more intuitive and appealing.

Bob Moore debugged my debugging algorithm while I was at SRI, and refuted my naive belief that a binary search technique could be readily used in the diagnosis of nontermination.

David Plaisted suggested an improvement to the algorithm that diagnoses termination with incorrect output, which I accepted without hesitation. He then refuted his improvement, but finally we made it together to the current divide-and-query algorithm.

Last but not least, the Tools group at Yale, and in particular John Ellis, Steve Wood, and Nat Mishkin, made the use of our DEC-20 and its surrounding machinery tolerable, and even fun. I cannot imagine what my productivity as a programmer and as a writer would have been without Z, SM, and the kind help they

provided me. They have also contributed to the literary aspect of the thesis, by telling me (or, more accurately, by allowing me to pick and put) the Zen parable about the drawing of the fish.