8062519

p 11
157
5

# NETL
# A SYSTEM FOR REPRESENTING AND USING REAL-WORLD KNOWLEDGE

SCOTT E. FAHLMAN

华南工学院
外国教材
中心图书室藏书

The MIT Press
Cambridge, Massachusetts, and London, England

E8052519

## PUBLISHER'S NOTE

This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. The time and expense of detailed editing and composition in print have been avoided by photographing the text of this book directly from the author's typescript.

Copyright © 1979 by
The Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form, or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

Printed in the United States of America.

Library of Congress Cataloging in Publication Data

Fahlman, Scott E.
NETL, a system for representing and using real-world knowledge.

Slightly edited version of the author's thesis, M.I.T., 1977.
Bibliography: p.
Includes index.
1. Artificial intelligence--Data processing.
2. NETL (Computer system) I. Title.
Q336.F33 001.53'5 78-10483
ISBN 0-262-0609-8

# NETL
# A System for Representing and Using Real-World Knowledge

## The MIT Press series in Artificial Intelligence

*Artificial Intelligence: An MIT Perspective, Volume I: Expert Problem Solving, Natural Language Understanding, Intelligent Computer Coaches, Representation and Learning* by Patrick Henry Winston and Richard Henry Brown, 1979

*Artificial Intelligence: An MIT perspective, Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation* by Patrick Henry Winston and Richard Henry Brown, 1979

*NETL: A System for Representing and Using Real-World Knowledge* by Scott E. Fahlman, 1979

*The Interpretation of Visual Motion* by Shimon Ullman, 1979

此为试读，需要完整PDF请访问：www.ertongbook.com

## SERIES FOREWORD

Artificial intelligence is the study of intelligence using the ideas and methods of computation. Unfortunately, a definition of intelligence seems impossible at the moment because intelligence appears to be an amalgam of so many information-processing and information-representation abilities.

Of course psychology, philosophy, linguistics, and related disciplines offer various perspectives and methodologies for studying intelligence. For the most part, however, the theories proposed in these fields are too incomplete and too vaguely stated to be realized in computational terms. Something more is needed, even though valuable ideas, relationships, and constraints can be gleaned from traditional studies of what are, after all, impressive existence proofs that intelligence is in fact possible.

Artificial intelligence offers a new perspective and a new methodology. Its central goal is to make computers intelligent, both to make them more useful and to understand the principles that make intelligence possible. That intelligent computers will be extremely useful is obvious. The more profound point is that artificial intelligence aims to understand intelligence using the ideas and methods of computation, thus offering a radically new and different basis for theory formation. Most of the people doing artificial intelligence believe that these theories will apply to any intelligent information processor, whether biological or solid state.

There are side effects that deserve attention, too. Any program that will successfully model even a small part of intelligence will be inherently massive and complex. Consequently, artificial intelligence continually confronts the limits of computer science technology. The problems encountered have been hard enough and interesting enough to seduce artificial intelligence people into working on them with enthusiasm. It is natural, then, that there has been a steady flow of ideas from artificial intelligence to computer science, and the flow shows no sign of abating.

The purpose of this MIT Press Series in Artificial Intelligence is to provide people in many areas, both professionals and students, with timely, detailed information about what is happening on the frontiers in research centers all over the world.

Patrick Henry Winston
Mike Brady

# PREFACE

The goal of the M.I.T. Press Series in Artificial Intelligence is to make available to a wide and interdisciplinary audience some of the work that is occuring at the frontiers of artificial intelligence research. In many cases, this means that the books in the series will be snapshots of work in progress, rather than the rounded, well-polished presentations that characterize older, more stable fields.

This book is such a snapshot. It is a slightly edited version of my Ph.D. thesis, completed in September of 1977 at the M.I.T. Artificial Intelligence Laboratory. It assumes that the reader is familiar in general with the principles of digital computers and has some appreciation for the difficulties involved in making these devices behave intelligently. It does not assume a knowledge of particular details or languages, except in the section on software simulation of the knowledge network and the appendices on hardware implementations. It is my hope that this book will be of interest to psychologists, linguists, and others not directly connected to the fields of computer science or artificial intelligence.

I would like to express my gratitude to the following people, who made essential contributions to this project:

To my thesis supervisor, Gerald Sussman, and my readers, Patrick Winston and Marvin Minsky, for their advice, support, and patience. I would also like to thank each for the special gifts that he brought to the task: Sussman, his contagious energy and enthusiasm; Winston, his broad perspective and good sense; and Minsky, his sense of direction.

To Drew McDermott, Bob Moore, Michael Genesereth, Mitch Marcus, and Jon Doyle for the many stimulating arguments and discussions that we have had about knowledge-representation and related topics.

To McDermott, Genesereth, Doyle, Candy Bullwinkle, Bill Martin, Guy Steele, Steve Rosenberg, Jim Stansfield, and

Johan de Kleer for reading drafts of the thesis and for making many useful comments and criticisms.

To Tom Knight, Jack Holloway, Kurt VanLehn, Ben Kuipers, David Marr, Carl Hewitt, Bob Woodham, Richard Stallman, Richard Greenblatt, Brian Smith, Chuck Rich, and Howie Shrobe for contributing good ideas, for pointing out things that I ought to read, for providing tools that were a pleasure to work with, for general moral support, and for combinations of the above.

To Bob Sjoberg and Brian Reid for their help in the area of digital calligraphy.

Finally, and most of all, I would like to thank my wife, Penny, for her support and understanding. This book is dedicated to her.

For two of the four years covered by this research, I was supported by an I.B.M. Graduate Fellowship. The laboratory was funded in large part by the Defense Advanced Research Projects Agency, as noted below.

Scott Fahlman

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in partby the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

8062519

## Table of Contents

此为试读，需要完整PDF请访问：www.ertongbook.com

8062519

*If a cat can kill a rat in a minute, how long would it be killing 60,000 rats? Ah, how long indeed! My private opinion is that the rats would kill the cat.*

*--- Lewis Carroll, on the advantages of parallelism*

# 1. Introduction

## 1.1 Preview

The human mind can do many remarkable things. Of these, perhaps the most remarkable is the mind's ability to store a huge quantity and variety of knowledge about its world, and to locate and retrieve whatever it needs from this storehouse at the proper time. This retrieval is very quick, very flexible, and in most cases seems almost effortless. If we are ever to create an artificial intelligence with human-like abilities, we will have to endow it with a comparable knowledge-handling facility; current knowledge-base systems fall far short of this goal. This report describes an approach to the problem of representing and using real-world knowledge in a computer.

The system presented here consists of two more-or-less independent parts. First, there is the system's parallel network memory scheme. Knowledge is stored as a pattern of interconnections of very simple parallel processing elements: node units which can store a dozen or so distinct marker-bits, and link units which can propagate these markers from node to node, in parallel through the network. Using these marker-bit movements, the parallel network system can perform searches and many common deductions very quickly: the time required is essentially constant, regardless of the size of the knowledge-base. The network is similar to the parallel marker-propagating network proposed by Quillian [1968, 1969], but is much more tightly controlled. This system is presented as a faster, more effective, and much simpler alternative to the currently popular approach of using domain-specific meta-knowledge, in the form of local procedures, to guide and limit serial searches in a large knowledge-base.

The second, more traditional part of the knowledge-base system is a vocabulary of conventions and processing algorithms -- in some sense, a language -- for representing various kinds of knowledge as nodes and links in the network. This set of conventions is called NETL. In many ways, NETL is similar to existing systems for representing real-world knowledge, such as the partitioned semantic networks of Hendrix [1975a, 1975b, 1976] and the frame-based KRL system [Borow & Winograd, 1976], but it differs from existing systems in three respects:

First, NETL incorporates a number of representational techniques -- new ideas and new combinations of old ideas -- which allow it to represent certain real-world concepts more precisely and more efficiently than earlier systems.

Second, NETL is built around a single, clear, explicit organizing concept: an effective knowledge-base system, in addition to storing and retrieving isolated facts, must provide the user with the ability to create and use *virtual copies* of descriptions stored in the memory. By "virtual copy", I mean that the knowledge-base system *behaves* as though a portion of the semantic network has been copied (with some specific alterations), but it does not actually create the redundant structure in memory. The descriptions that are copied in this way may be arbitrarily large and complex, with parts, sub-parts, and internal relationships. This entire structure is inherited by the copy, not just a few global properties. NETL is not unique in *providing* such a virtual copy facility, but it is unique in *stating* this goal explicitly, in clear and simple terms, and in relating all of its representational machinery to this goal. This adds considerably to the conceptual clarity and unity of the resulting system, and it provides us with a way of determining whether the system's accessing mechanisms do what they are supposed to do.

Finally, NETL has been designed to operate efficiently on the parallel network machine described above, and to exploit this machine's special abilities. Most of the ideas in NETL are applicable to knowledge-base systems on serial machines as well.

A simulator for the parallel network system has been implemented in MACLISP, and an experimental version of NETL is running on this simulator. A number of test-case results and simulated timings will be presented.

## 1.2 The Knowledge-Base Problem

Suppose I tell you that a certain animal -- let's call him Clyde -- is an elephant. You accept this simple assertion and file it away with no apparent display of mental effort. And yet, as a result of this transaction, you suddenly appear to know a great deal about Clyde. You can tell me, with a fair degree of certainty, how many legs he has, what color he is, and whether he would be a good pet in a small third-floor apartment. You know not only that he has eyes, but what they are used for, and what it implies if they are closed. If I try to tell you that Clyde builds his nest in a tree or that he is a virtuoso on the piano or that he amuses himself by hiding in a teacup, you will immediately begin to doubt my credibility. And you can do this very quickly and easily, with none of the sort of apparent mental effort that would accompany, say, adding two four-digit numbers. This effortlessness may be an illusion, but it is a compelling one.

"Elephant", of course, is not the only concept that behaves in this way. The average person knows a huge number of concepts of comparable or greater complexity -- the number is probably in the millions. Consider for a moment the layers of structure and meaning that are attached to concepts like lawsuit, birthday party, fire, mother, walrus, cabbage, or king. These are words we use casually in our daily lives, and yet each of them represents a very substantial package of information. In technical fields (except, perhaps, for the more austere parts of mathematics) the situation is the same. Consider how much you would have to tell someone in order to fully convey the meaning of concepts like meson, local oscillator, hash-table, valence, ribosome, or leukemia. And yet, once these concepts are built up, they can be tossed around with abandon and can be used as the building blocks for concepts of even greater complexity.

The point is not just that we can handle large chunks of knowledge as though they were atoms; the important thing is that we can find our way through these complex, nested structures to whatever individual fact or relationship we might need at any given time, that we can do this in a very flexible and efficient way, and that we can somehow avoid having to look individually at each of the vast number of facts that could be -- but are not -- relevant to the problem at hand. If I tell you that a house burned down, and that the fire started at a child's birthday party, you will think immediately of the candles on the cake and perhaps of the many paper decorations. You will not, in all probability, find yourself thinking about playing pin-the-tail-on-the-donkey or about the color of the cake's icing or about the fact that birthdays come once a year. These concepts are there when you need them, but they do not seem to slow down the search for a link between fires and birthday parties. If, hidden away somewhere, there is a sequential search for this connection, that search is remarkably quick and efficient, and it does not become noticeably slower as the knowledge base expands to its adult proportions.

This impressive ability to store and access a large and diverse body of knowledge is a central feature of human intelligence. The knowledge-base system provides essential support for the other components of intelligence: the peripheral processes that handle such things as vision and speech understanding, and the linear, sequential, conscious kinds of thinking that characterize our problem-solving behavior. The knowledge base is the common ground for these diverse elements, the glue that holds everything else together.

It follows, then, that any *artificial* intelligence, if it is to be even remotely human-like in its capabilities, must include a knowledge-base system with abilities comparable to those possessed by humans. To date, in the field of AI research, we have been unable to achieve or even approach this goal. We can make -- and have made -- a certain amount of progress toward understanding the sensory and problem-solving areas of thought by confining our investigations to relatively knowledge-free problem domains -- tasks like cryptarithmetic puzzles and the symbolic integration of mathematical expressions. We can make still more progress by patching together tiny knowledge bases, just sufficient to serve as scaffolding for whatever test cases we are working on at the moment. But until we can find an adequate solution to the knowledge-base problem, all of our work will be fragmented and somewhat distorted. Sooner or later, we will have to confront that elephant.

The problem is not that we are unable to store and retrieve enough *explicit* knowledge -- that problem was solved long ago. In the property lists of LISP, in the hash-tables of LEAP and SAIL [Feldman & Rovner, 1969], and in the indexing structures of the PLANNER-related languages [Hewitt, 1972; Sussman, Charniak, Winograd, 1971; McDermott & Sussman, 1972], we can store away an arbitrarily large body of assertions and can easily retrieve any one of these later with a matching query. But the key word here is "matching": the datum to be found must be explicitly present, and it must be in the proper format for the match to succeed. These systems (ignoring, for a moment, their procedural components) give us no direct access to the much larger body of information that is *implicit* in the set of facts at hand. If we know things about "every elephant" or "every animal" or "every object bigger than a breadbox" and the questions are about Clyde, we need some way to connect the question to the answer. That means deduction, and deduction means search. To be sure that it has found all of the information relevant to Clyde, a knowledge-base system would have to examine a potentially very large set of related concepts.

The problem, then, is to find a way to perform this search in a reasonable time, even when the data base is huge. We can perform the deductions in antecedent fashion as new facts are added; we can perform them in consequent fashion in response to specific queries; or we can use some combination of these approaches, but the problem remains basically the same: our current search techniques are much too slow to handle a knowledge-base of sufficient size to produce a human-like intelligence, even in a restricted problem-domain.

Note that I am not referring here to the difficult deductions that people perform consciously: solving puzzles,

此为试读，需要完整PDF请访问：www.ertongbook.com