Chapter 11. Knowledge Representation and Reasoning

The Quest for Artificial Intelligence, Nilsson, N. J., 2009.

Lecture Notes on Artificial Intelligence

Summarized by Ha, Jung-Woo and Lee, Beom-Jin

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National Univertisy

http://bi.snu.ac.kr

Contents

Overview

- Methods for knowledge representation and reasoning from Mid-1960s and Mid-1970s
  - Symbolic logic and its deductions
    - Predicate calculus
    - For proving theories
  - Situation calculus
  - Logic programming: PROLOG
  - Sematic networks: HAM, MEMS, MENTAL
  - Script and Frames

Introduction

- Knowledge
  - For intelligent system
  - The mean to draw conclusion from or act on
- Knowledge representation
  - Procedural
    - Coordinate and control the specific action (ex. hitting a tennis ball)
    - Programs using the knowledge
    - Specific task program
  - Declarative
    - Declarative sentence (I am a 25 years old)
    - Symbolic structures
    - General task program

Chapter 11. Knowledge Representation and Reasoning

11.1 Deductions in Symbolic Logic

Deductions in Symbolic Logic

- The predicate calculus
  - From Aristotle to G. Boole and McCarthy
  - Ex. Aristotle syllogism
    - 1. $(\forall x)[Man(x) \supset Mortal(x)]$

      (The expression “$(\forall x)$” is a way of writing “for all x”; and the expression “$\supset$” is a way of writing "implies that." “Man(x)” is a way of writing “x is a man”; and “Mortal(x)” is a way of writing “x is mortal.” Thus, the entire expression is a way of writing “for all x, x is a man implies that x is mortal” or, equivalently, “all men are mortal.”)
    - 2. Man(Socrates) (Socrates is a man.)
    - 3. Therefore, Mortal(Socrates) (Socrates is mortal.)
  - “Therefore,” is an example of a *deduction*
  - Rules of inference (ex. Modus ponens)

Deductions in Symbolic Logic

- Early works on deduction in symbolic logic
  - Programs using inference rule (1960s) for proving theorems in the predicate calculus
    - P. Gilmore, H. Wang, and D. Prawitz (IBM)
    - F. Black (Harvard)
  - QA3 (Question Answering)
    - C. C. Green implemented a new deduction method developed by J. A. Robinson
      - From two other statements, a new statement is generated by rules (ex. $P \vee \neg Q$ and P produces Q)
      - Key contribution: how resolution could be applied to general expressions in the predicate calculus
    - Example
- So just as with programs for playing games, LT, and proving geometry theorems, deduction programs need to try many possibilities in their search for a solution

Chapter 11. Knowledge Representation and Reasoning

11.2 The Situation Calculus

The Situation Calculus

- Situation calculus
  - Where one could write logical statements that explicitly named the situation in which something or other was true
  - Ex. “What is a program for rearranging a list of numbers so that they are in increasing numerical order?”
  - Block case
    - block A is on top of block B in some situation S
      $\rightarrow$ On(A, B, S)
    - block A is blue in all situations
      $\rightarrow (\forall s)\text{Blue}(A, s)$
    - there exists some situation in which block A is on block B
      $\rightarrow (\exists s)\text{On}(A, B, s)$
  - QA3 can deduce situation calculus → robot plan

Chapter 11. Knowledge Representation and Reasoning

11.3 Logic Programming

Logic Programming

- Green's automatic programming
  - QA3 can construct simple computer programs
  - The first attempt to write programs using logical statements
- SL-resolution : A. Kowalski and D. Kuehner
- PROLOG (1972)
  - A. Comerauer, P. Roussel, and A. Kowalski
  - An ordered sequence of logical statements
  - The exact order in which these statements are written, along with some other constructs, is the key to efficient program execution

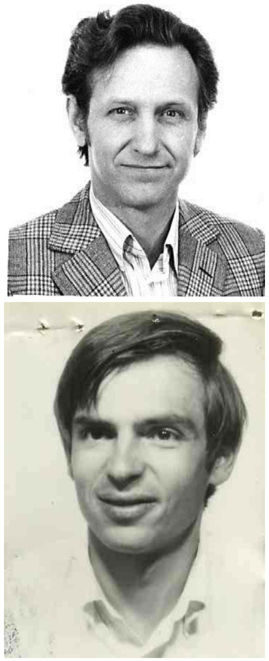

Figure 11.1: Robert Kowalski (top) and Alain Colmerauer (bottom)

Chapter 11. Knowledge Representation and Reasoning

11.4 Semantic Networks

Semantic Networks

- Semantic networks
  - Another format for representing declarative knowledge
- Human Associative Memory (HAM)
  - G. Bower and J. Anderson (1970s)
  - Network-based human memory
  - Parse simple propositional sentences and store them in the semantic network structure
  - With accumulated memory, HAM can answer simple questions
- MEMS and MENTAL: S. C. Shapiro (1971)
  - MEMS: a network structure for storing semantic information
  - MENTAL: aided MEMS in deducing new information from that already stored
- SNePS: S. C. Shapiro
  - Combination of logical representation with those of network representations used for natural language understanding

Semantic Networks

- Conceptual dependency representations for natural language sentences
  - R. C. Schank
  - People transform natural language sentences into "conceptual structures independent of the particular language where the sentences were expressed.

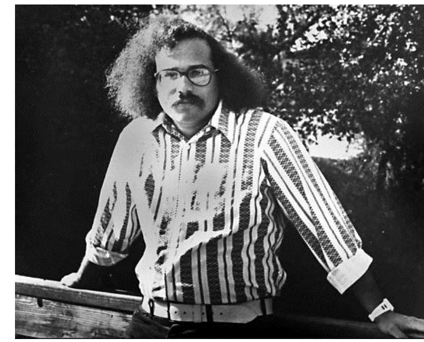

Figure 11.2: Roger Schank.

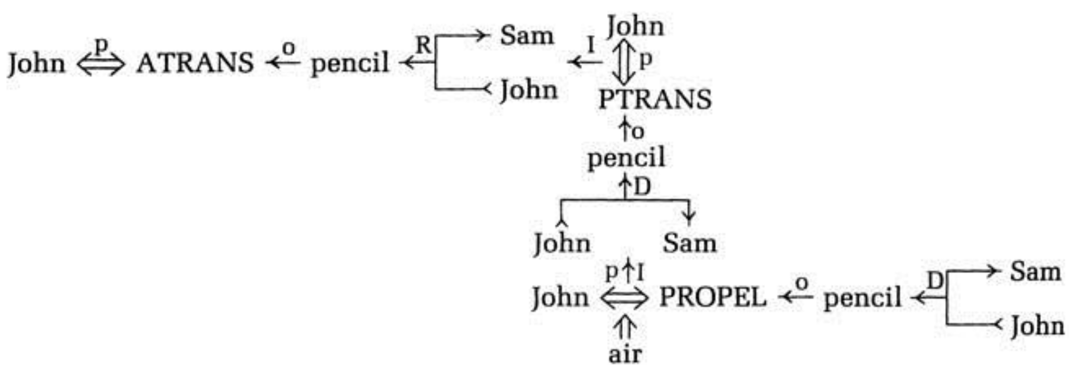

Figure 11.3: Conceptual structure for "John threw the pencil to Sam." (From Roger C. Schank, "Identication of Conceptualizations Underlying Natural Langauge," in Roger Schank and Kenneth Colby (eds.), *Computer Models of Thought and Language*, p. 226, San Francisco: W. H. Freeman and Co., 1973.)

Chapter 11. Knowledge Representation and Reasoning

11.5 Scripts and Frames

Scripts and Frames

- Graphical knowledge representations
  - Semantic networks and conceptual structures
  - Efficient computationally due to participating in the same chain of reasoning
- Scripts
  - Proposed by R. Schank and R. Abelson
  - A script is a way of representing what they call “specific knowledge – detailed knowledge about a situation or event that “we have been through many times.”
  - Example
- Frames
  - Proposed by M. Minsky
  - a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party.
  - Implementation: FRL and KRL

Chapter 11. Knowledge Representation and Reasoning

Appendix

Deductions in Symbolic Logic

- QA3
  - Resolution-based deduction system
  - The advantage of resolution
    - Implemented in programs to make deductions from a set of logical statements consisting of "clauses"
  - Ex.
    - 1. ROBOT(Rob) (Rob is a robot.)
    - 2. $(\forall x)[\mathrm{MACHINE}(x) \supset \neg \mathrm{ANIMAL}(x)]$

      (x is a machine implies that it is not an animal.)

      The system is then asked "Is everything an animal?" by having it attempt to deduce the statement
    - 3. $(\forall x)\mathrm{ANIMAL}(x)$

      QA3 answers "NO" and gives a "counterexample"
    - 4. x = Rob

      (This indicates that :ANIMAL(Rob) contradicts what was to be deduced.)

Scripts and Frames

- An example of scripts

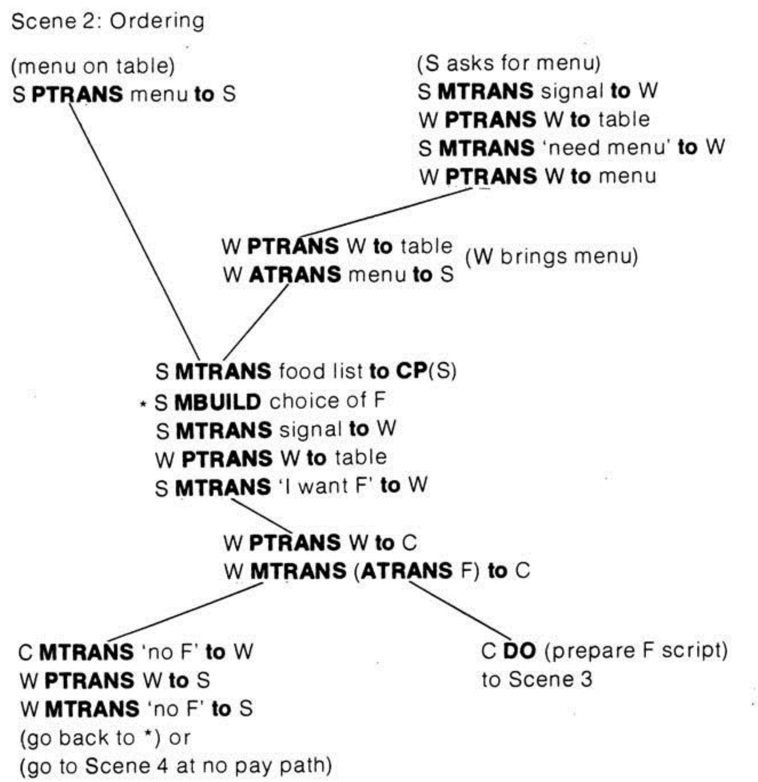

Figure 11.4: A scene in the restaurant script. (From Roger C. Schank and Robert P. Abelson, *Scripts*, *Plans*, *Goals, and Understanding: An Inquiry into Human Knowledge Structures*, p. 43, Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.)