Logic Programming
Introduction

Michael Genesereth
Computer Science Department
Stanford University

Lecture will begin at ~1:35 PDT.

Logic Programming (Spoiler Alert)

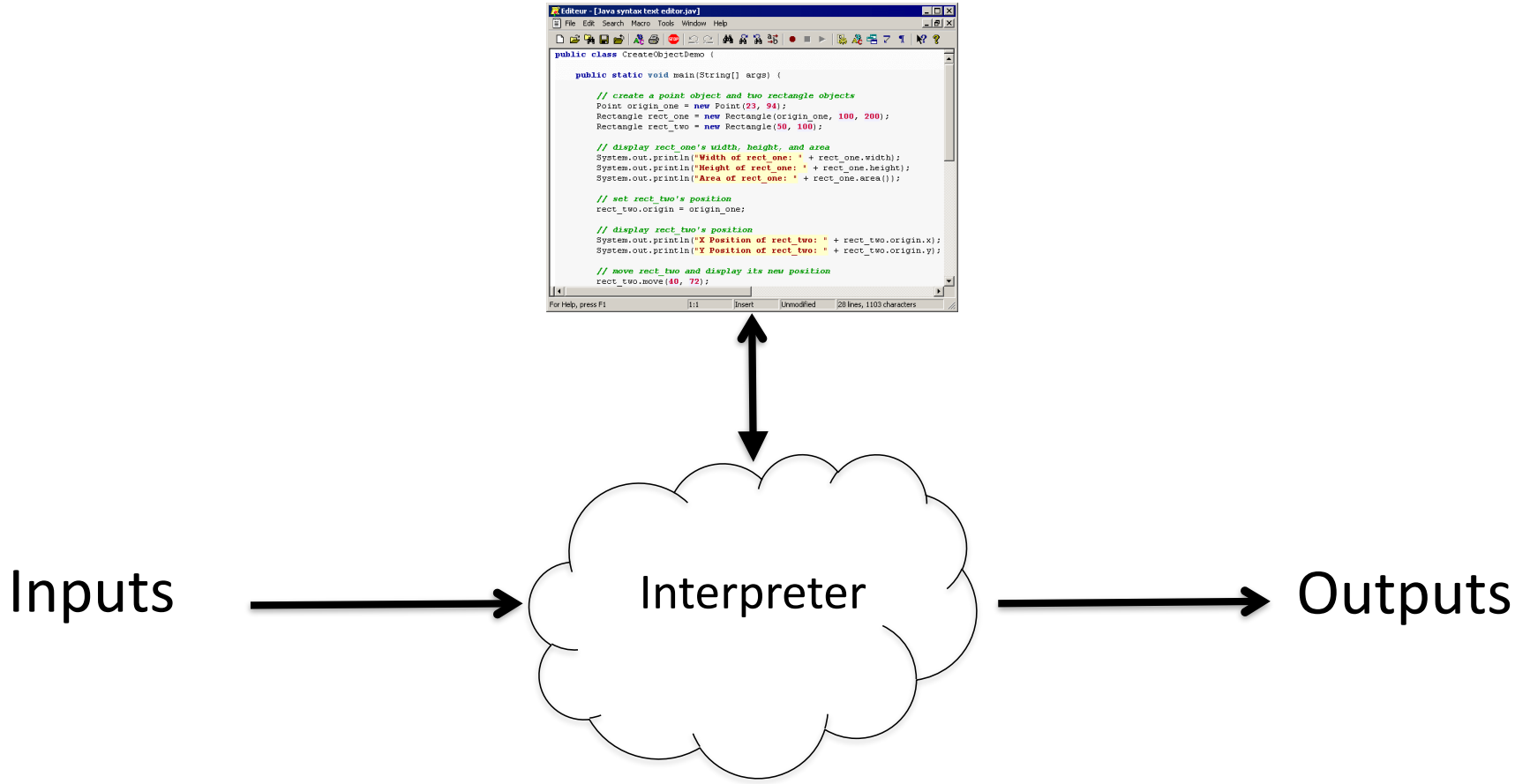
Logic Programming is a style of programming based on Symbolic Logic.

Logic Program is a collection of sentences encoded in the language of Symbolic Logic.

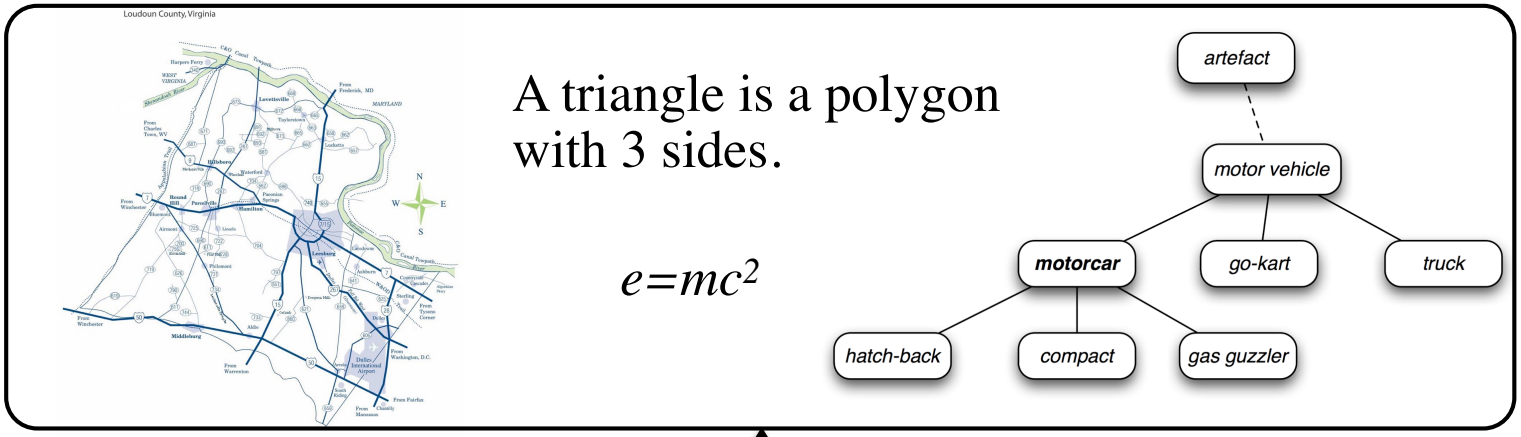
Logic Programming Language is a specific language for writing such programs.

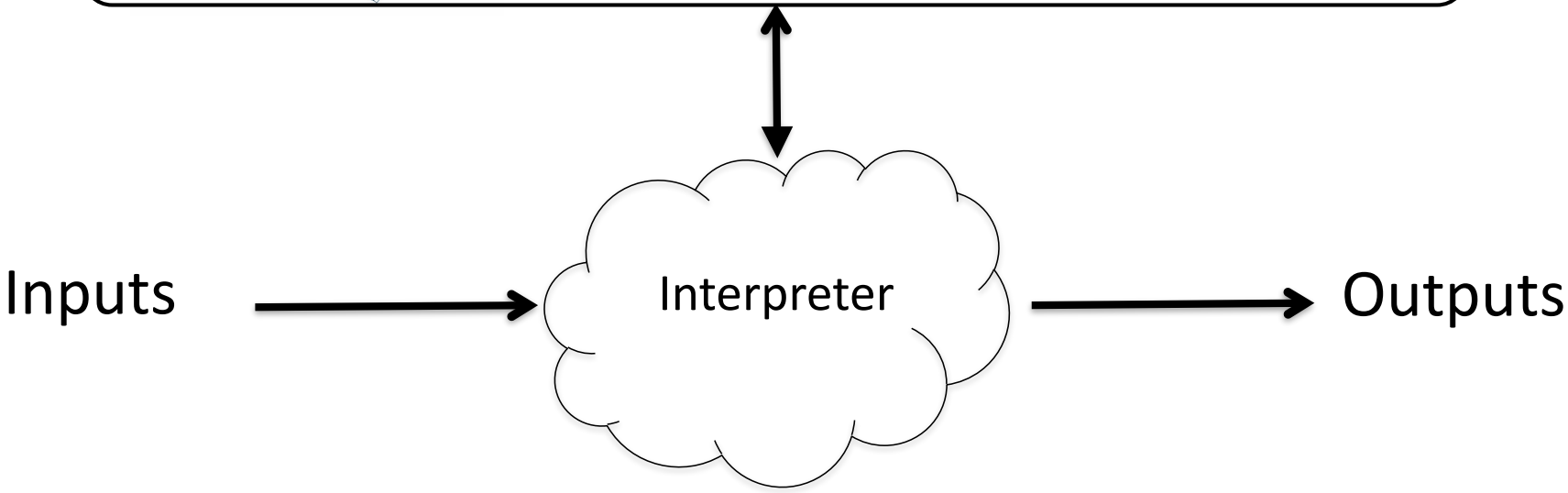
Logic Programming System is a computer system that manages the creation, modification, and execution of logic programs.

Imperative Programming



Declarative Programming





Runnable Specifications

Specification

What we believe about the *application area*
What we want to know or to achieve in *application area*

With no arbitrary decisions
With no concern for internal processing details

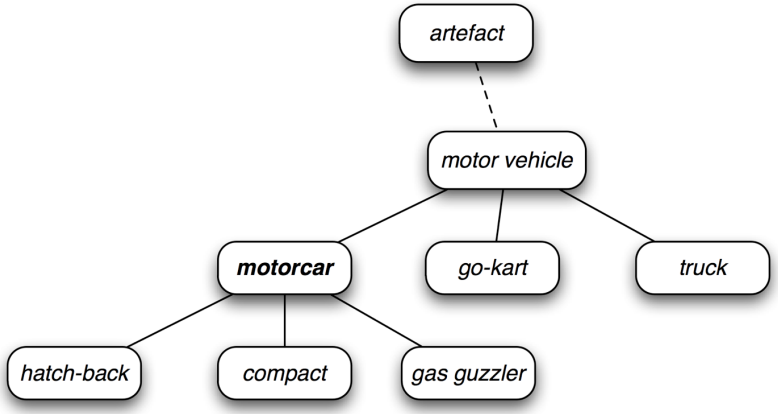
Runnable

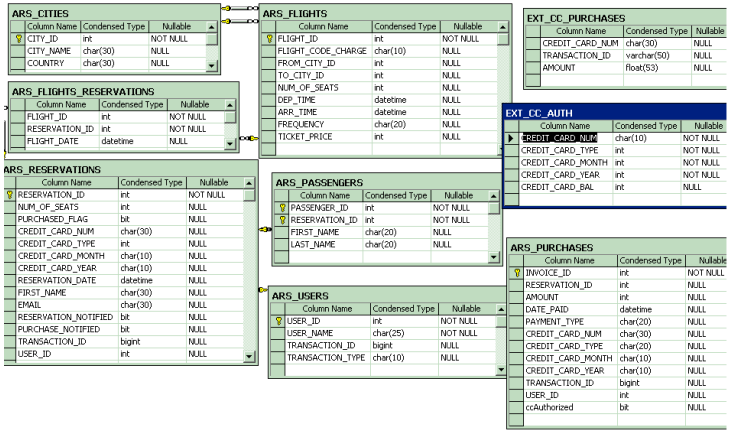
Can be directly **interpreted**
Can be **compiled** into traditional programs

Runnable Specifications

A logic program
is a
runnable specification.

Specialized Languages





A triangle is a polygon with 3 sides.

Logic as a Specification Language

Language

Declarative language

+

Highly expressive

Other declarative languages exist but statements in most of those languages can be translated to logical form.

Interpreter

Automated Reasoners capable of drawing conclusions
Can take advantage of domain-dependent reasoners
but are also capable domain-independent reasoning

Benefits

Programming Ease

Easier to create and modify than traditional programs

Programmers can get by with **little or no knowledge** of the capabilities of systems executing those programs.

Less work. The specification is the program; no need to make choices about data structures and algorithms.

Easier to learn logic programming than traditional programming. Think spreadsheets.

Oddly, expert computer programmers often have more trouble with logic programming than novices.

Agility

Ability to respond to changing circumstances or goals

Versatility

Ability to be used for multiple purposes

Sample Program

A person X is the grandparent of a person Z if and only if there is a person Y such that X is the parent of Y and Y is the parent of Z.

Uses

Determine whether Art is the grandparent of Cal.
Determine all of the grandchildren of Art.
Compute the grandparents of Cal.
Compute all grandparent-grandchildren pairs.

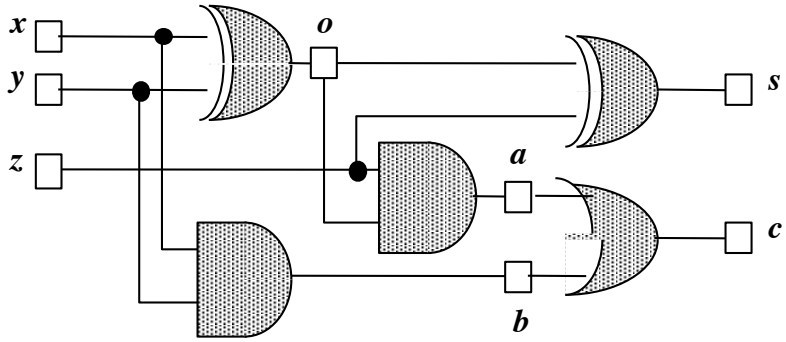
McCarthy's Example of Versatility

McCarthy's Example of Versatility

Successes

Engineering

Circuit:



Premises:

$$o \Leftrightarrow (x \wedge \neg y) \vee (\neg x \wedge y)$$

$$a \Leftrightarrow z \wedge o$$

$$b \Leftrightarrow x \wedge y$$

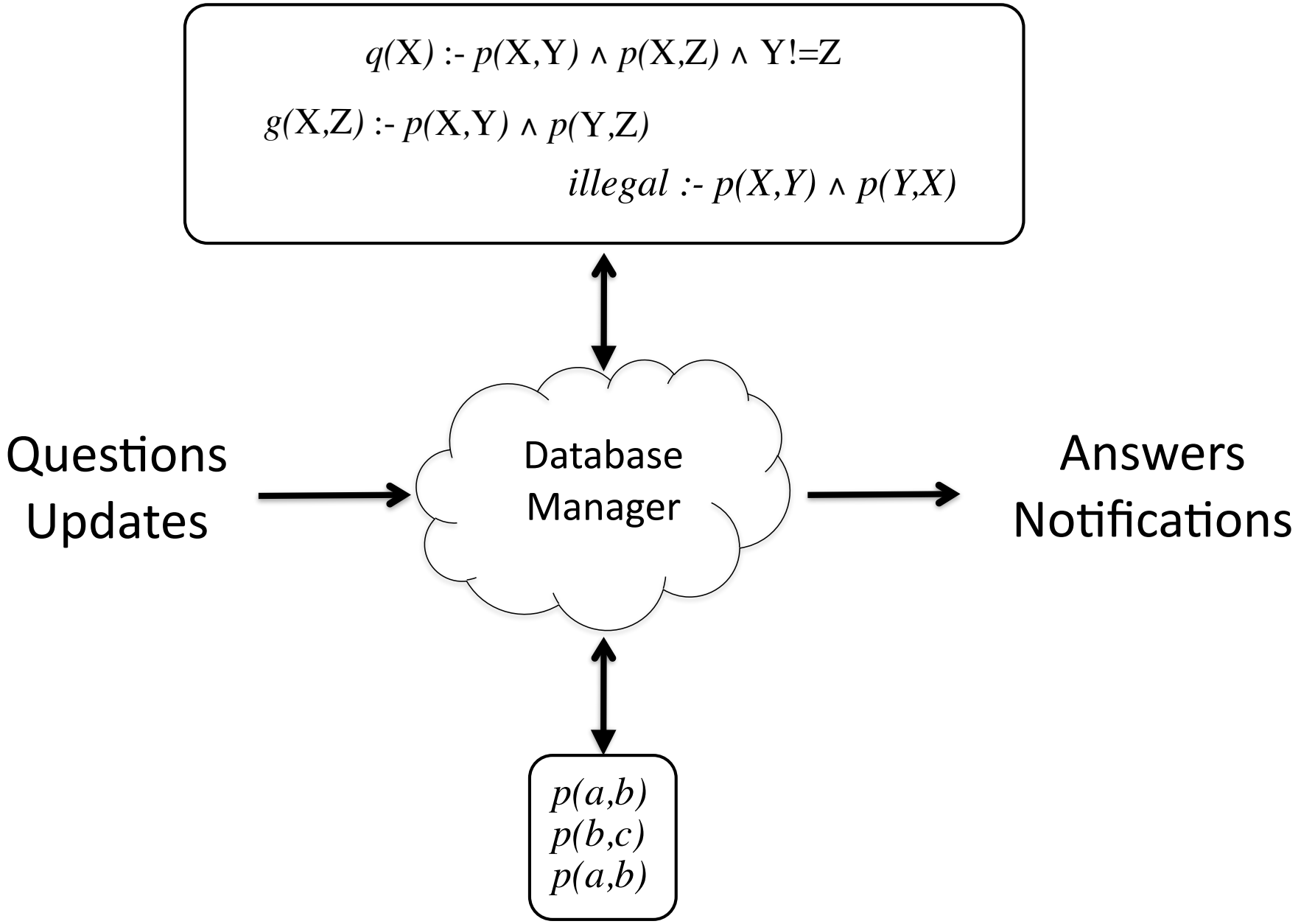
$$s \Leftrightarrow (o \wedge \neg z) \vee (\neg o \wedge z)$$

$$c \Leftrightarrow a \vee b$$

Applications:

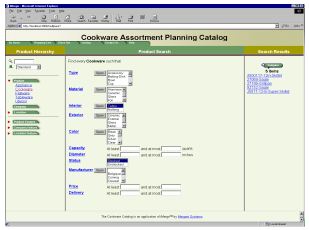
- Simulation
- Configuration
- Diagnosis
- Test Generation

Deductive Databases



Data Integration

Integrated Search



Side-by-side Comparison

Integration Engine

Supplier 1

Supplier 2

Supplier 3

Supplier 4

Manufacturer 1

Manufacturer 2

Product analysis

Satisfaction Ratings

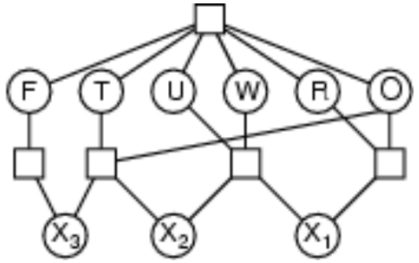
Marketplace Data

Constraint Satisfaction

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

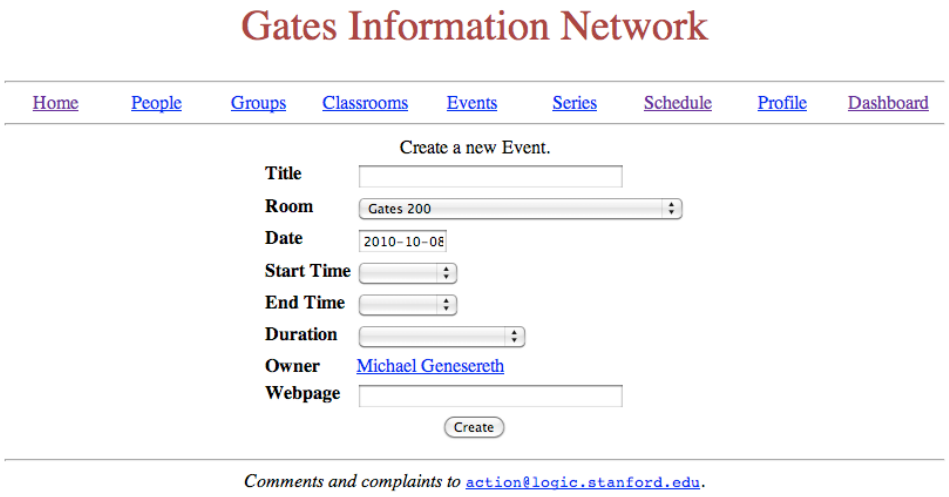
Example: Cryptarithmetic

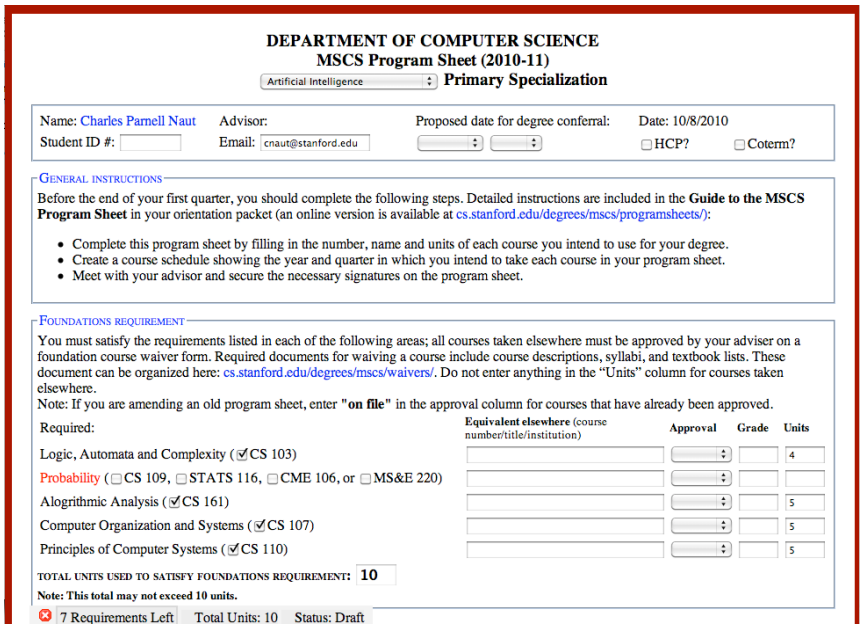
```
  T W O
+ T W O
-------
F O U R
```

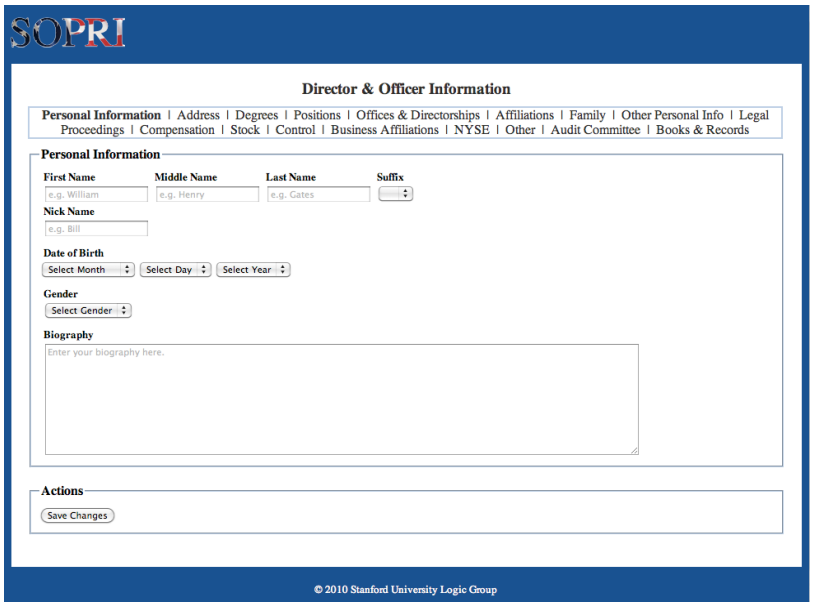


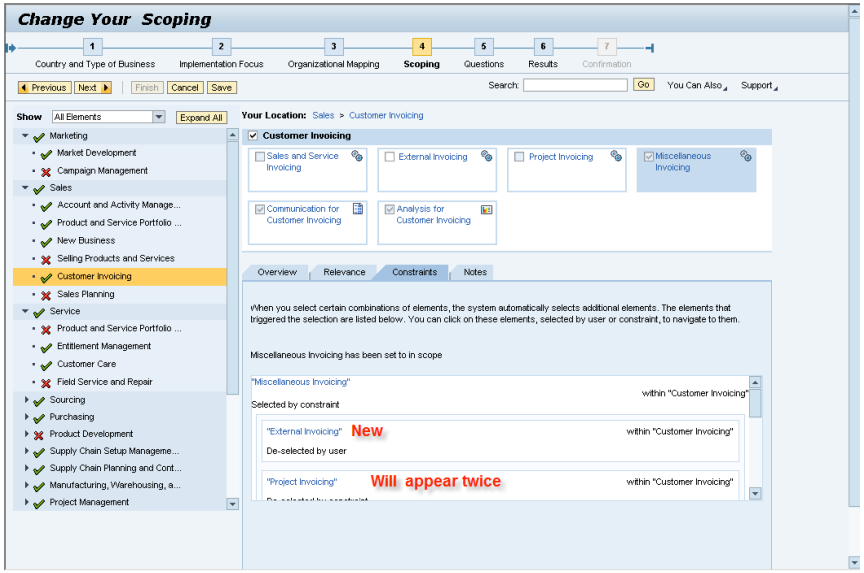
- Variables: *F T U W R O* $X_1\ X_2\ X_3$
- Domains: {*0,1,2,3,4,5,6,7,8,9*} {0,1}
- Constraints: *Alldiff (F,T,U,W,R,O)*
  - $O + O = R + 10 \cdot X_1$
  - $X_1 + W + W = U + 10 \cdot X_2$
  - $X_2 + T + T = O + 10 \cdot X_3$
  - $X_3 = F, T \neq 0, F \neq 0$

Worksheets









http://logicprogramming.stanford.edu/examples/programsheets/demonstration.html

Business Rules

SAP®

ORACLE®

IBM®

Computational Law

Computational Law is that branch of legal informatics concerned with the mechanization of legal reasoning.

Automated Compliance Management

Legal analysis of specific cases

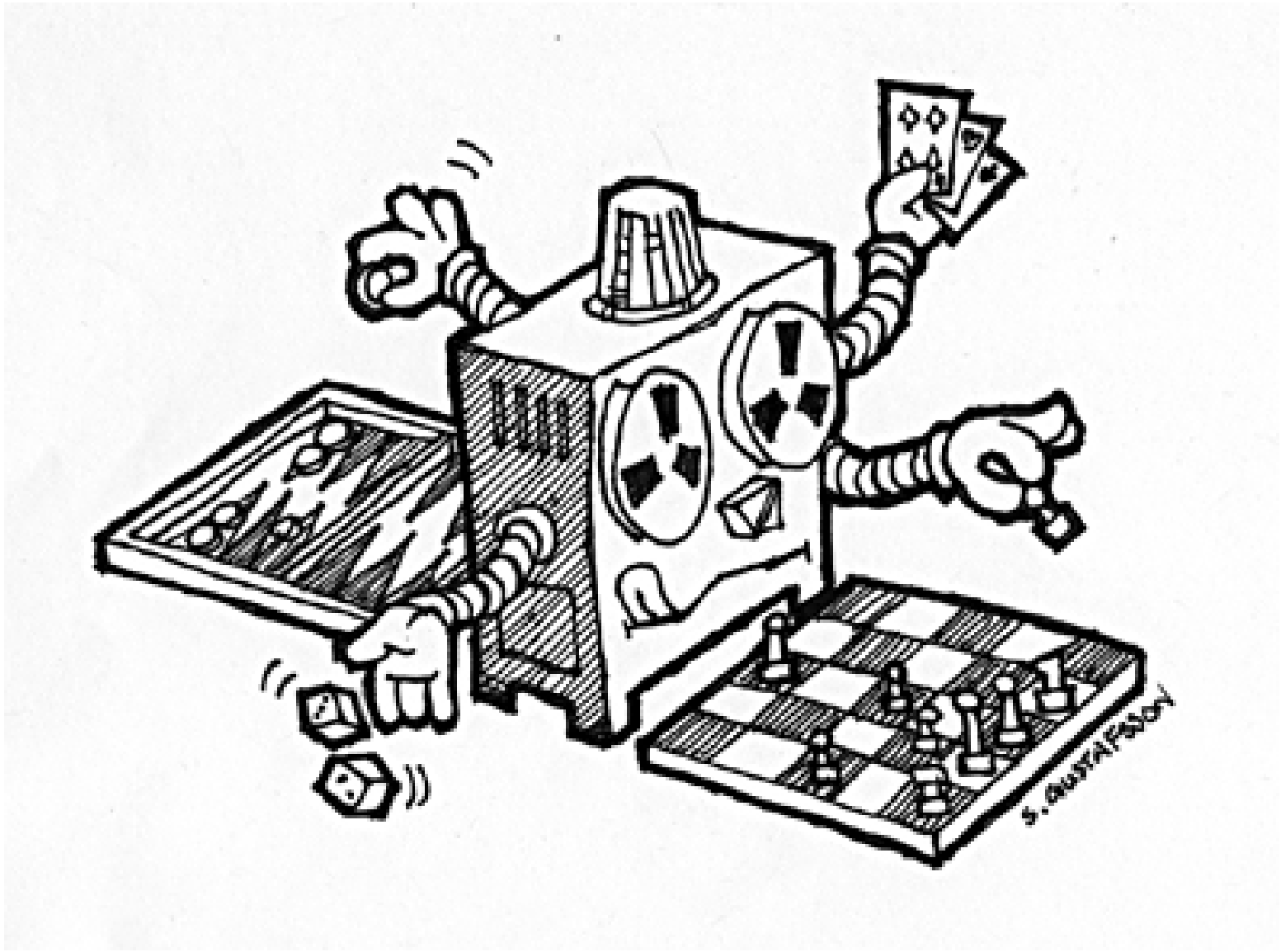
Planning for compliance in specific cases

Analysis of regulations for overlap, consistency, etc.

http://logicprogramming.stanford.edu/examples/portico/demonstration.html

General Game Playing

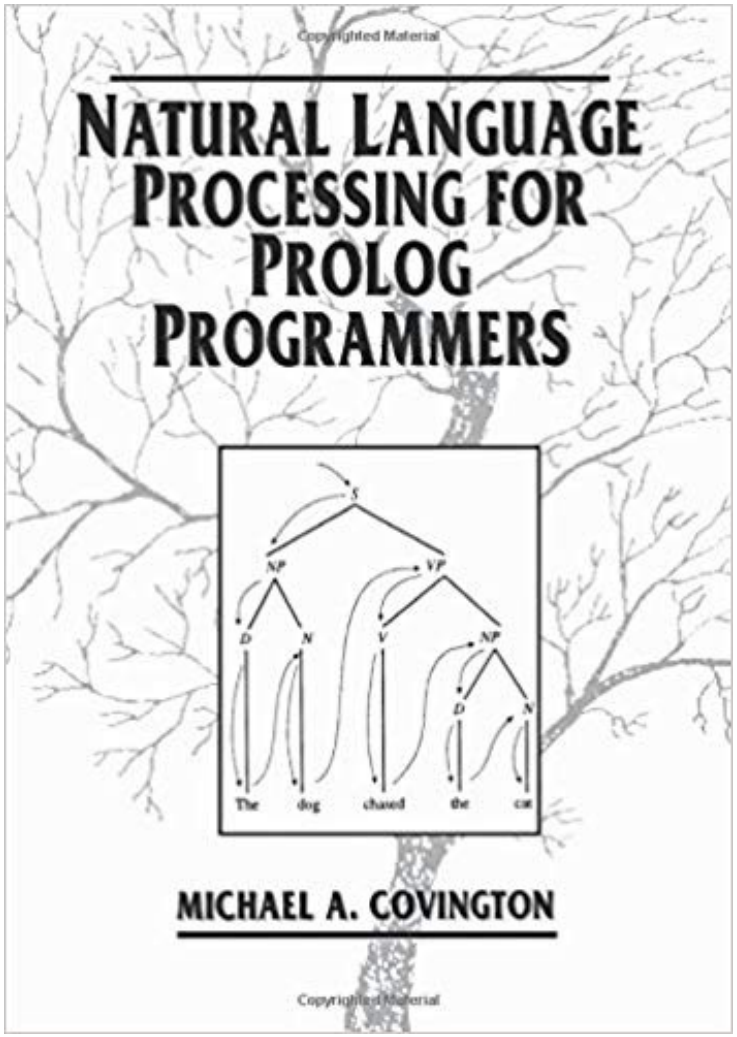
General Game Playing

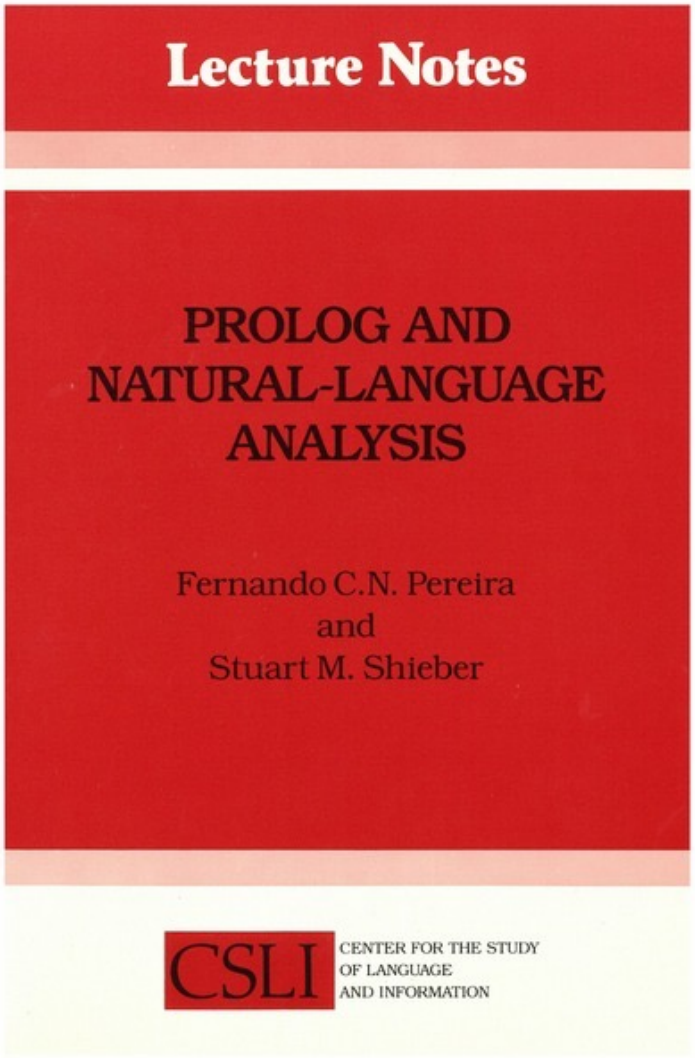


http://logicprogramming.stanford.edu/examples/nineboard/demonstration.html

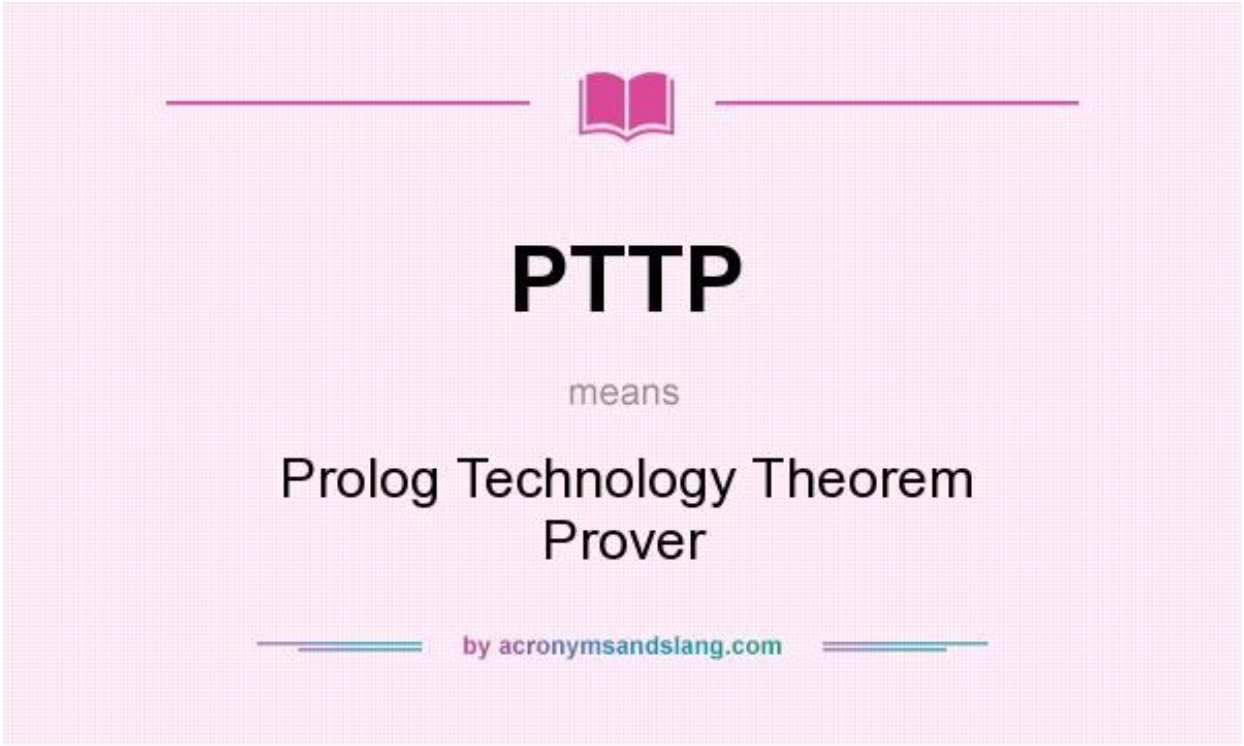
Non-Successes

Natural Language Processing

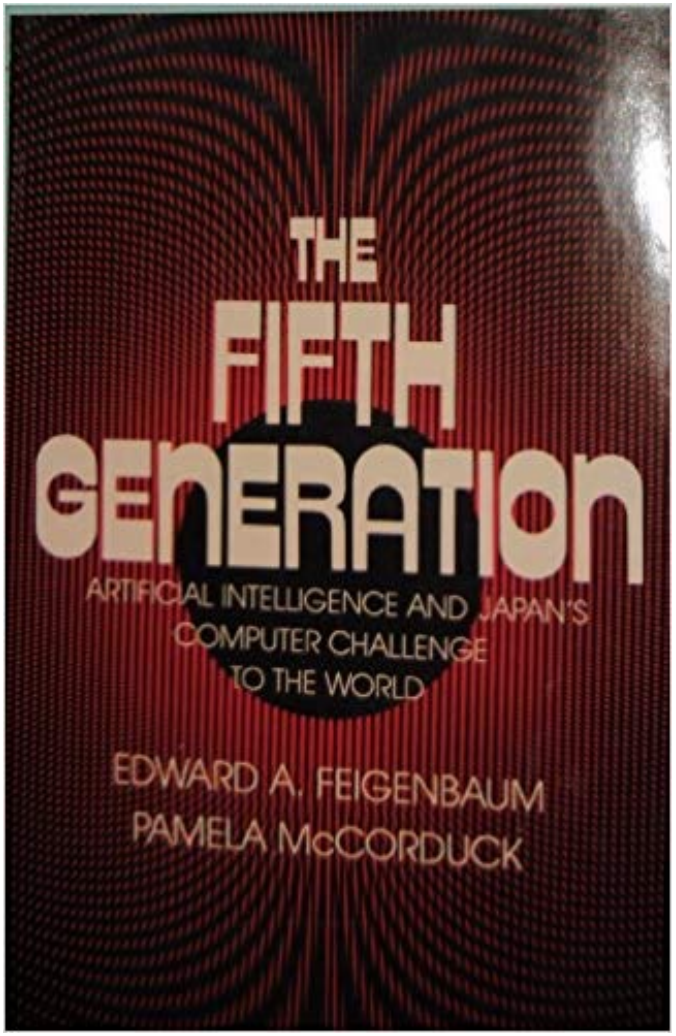




Theorem Proving

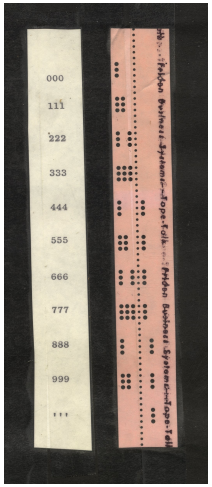


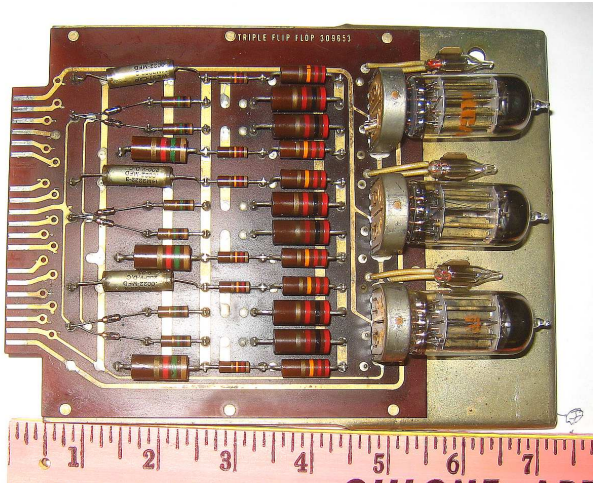
Japan's Fifth Generation Project

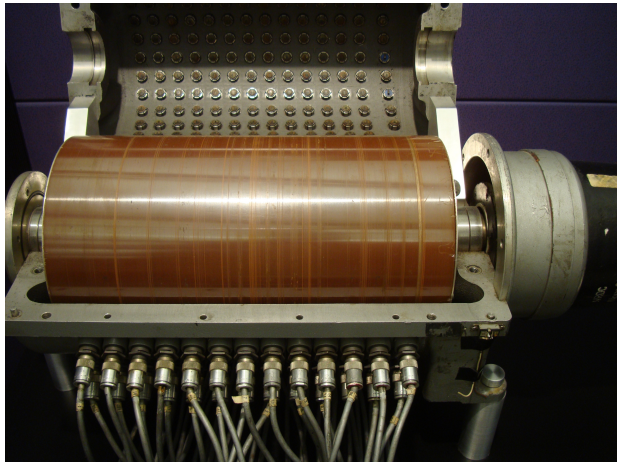


History

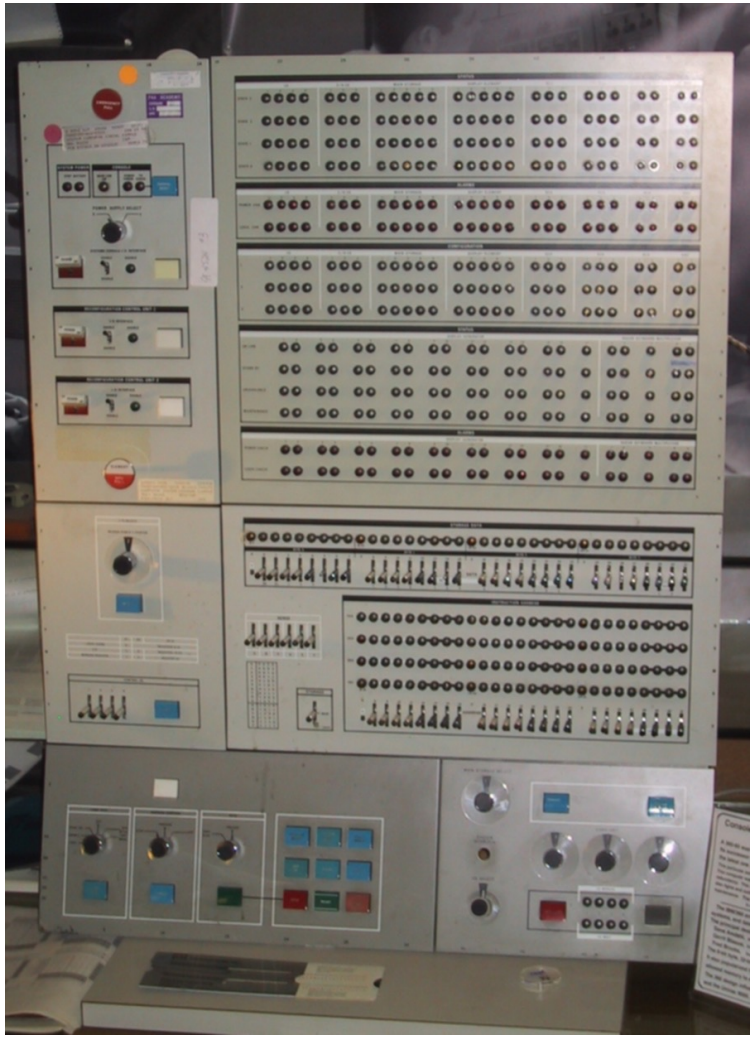
LGP-30


000
111
222
333
444
555
666
777
888
999


TRIPLE FLIP FLOP 300653



IBM 360



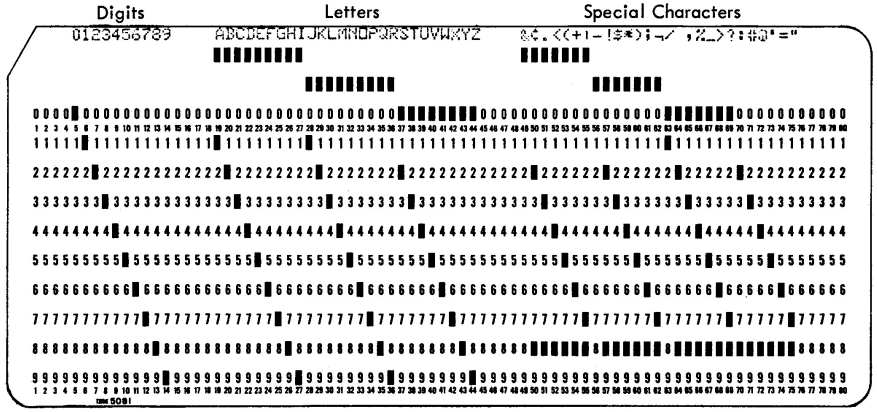
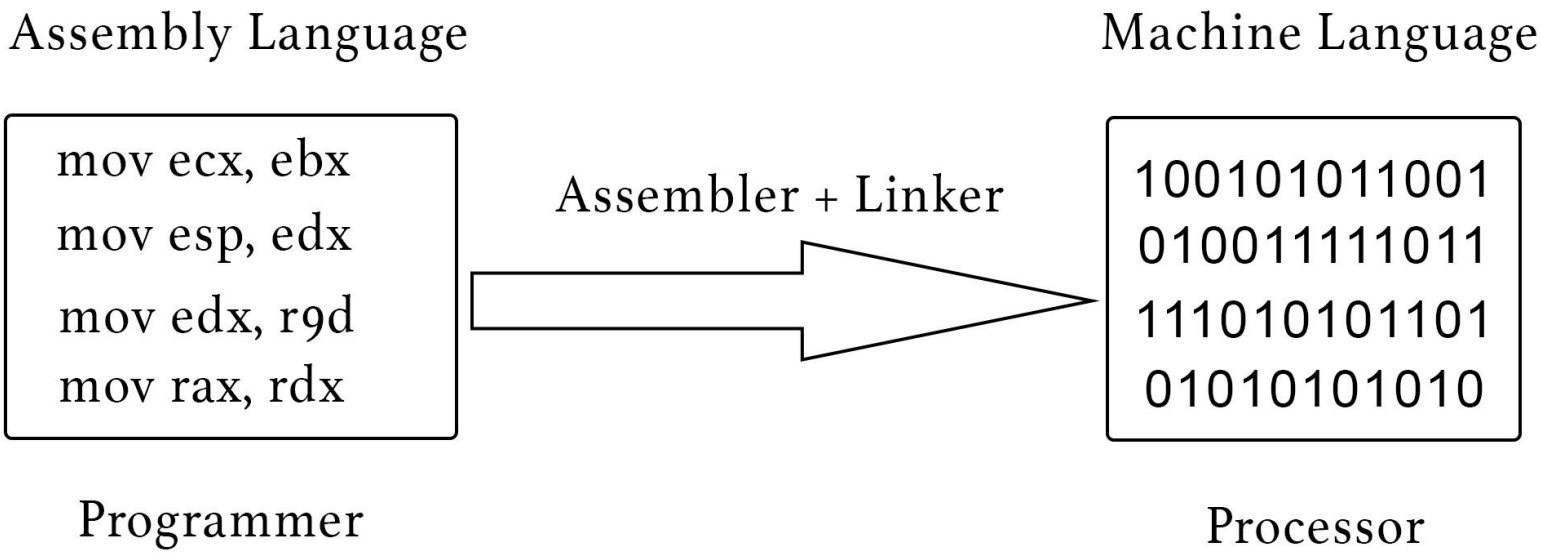
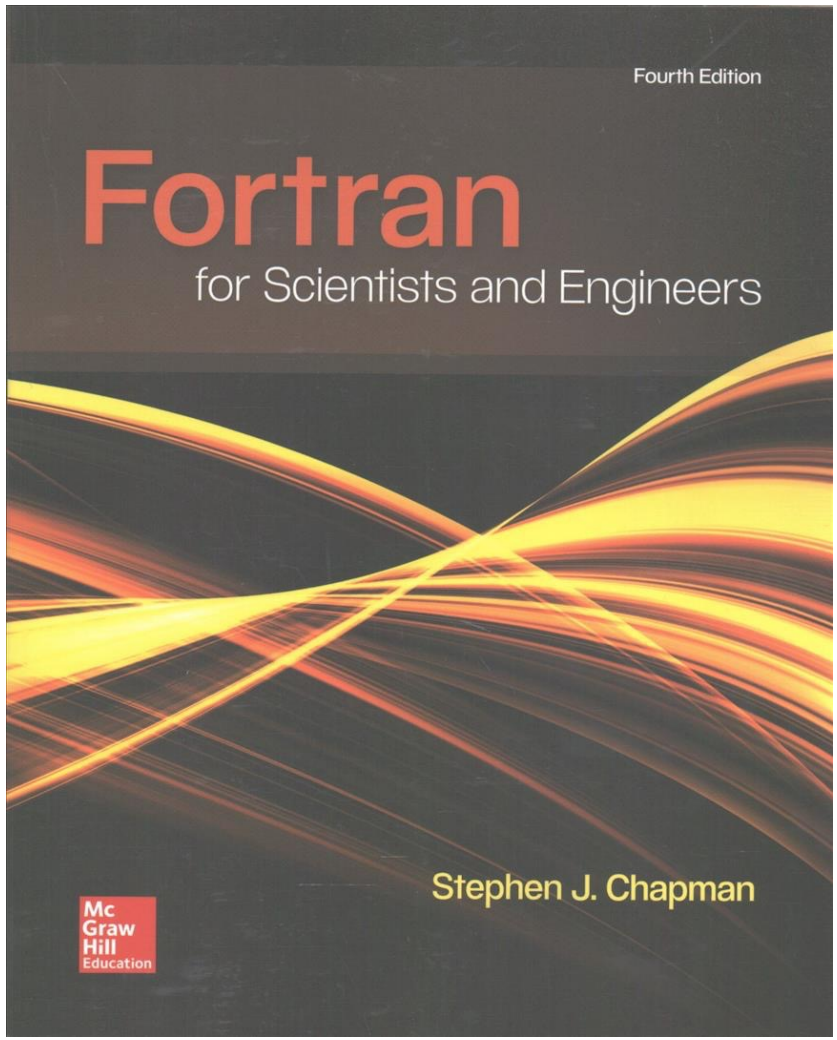


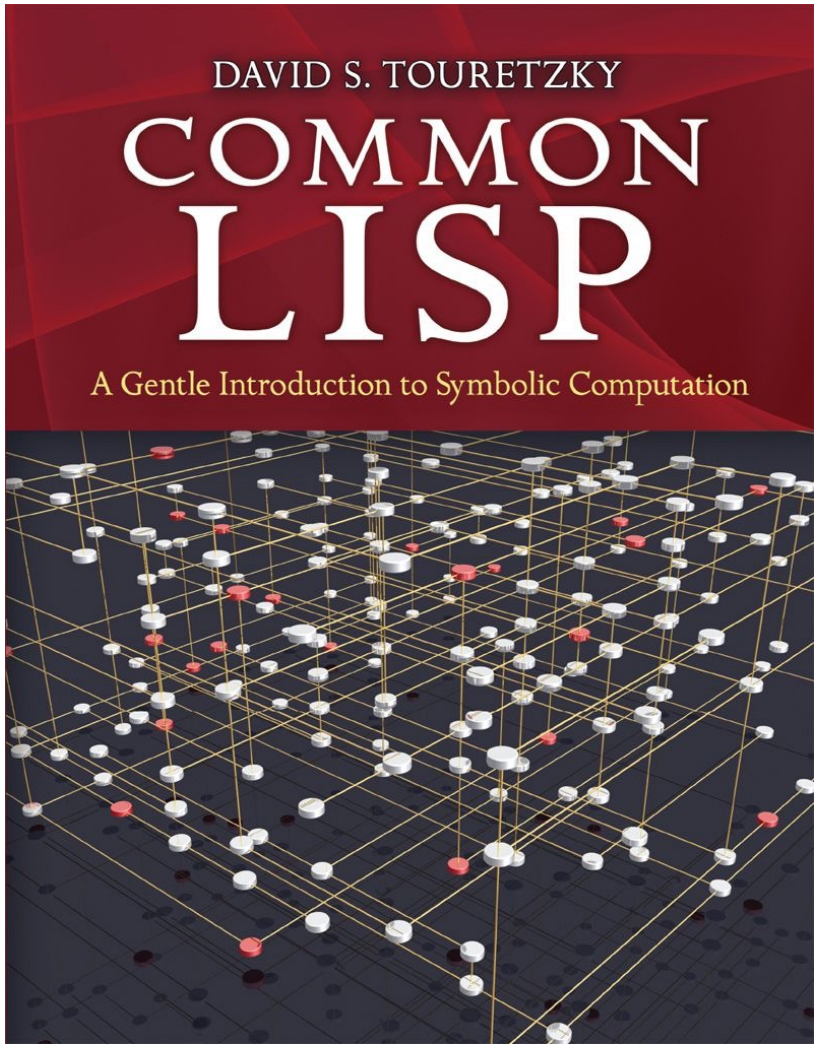
Figure 4. Card Codes and Graphics for 64-Character Set

Assembly Language

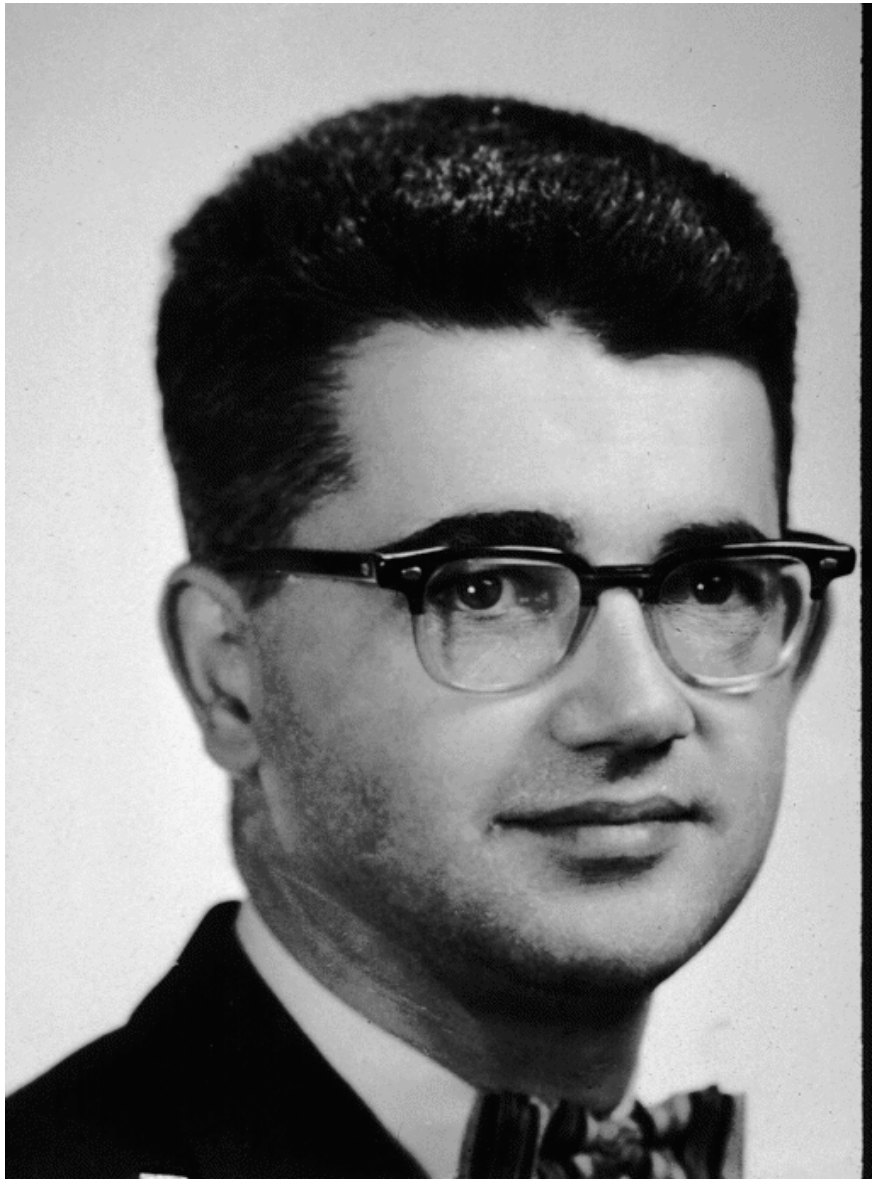


Higher Level Languages





John McCarthy



*The main advantage we expect the **advice taker** to have is that its behavior will be improvable merely by making statements to it, telling it about its … environment and what is wanted from it.*

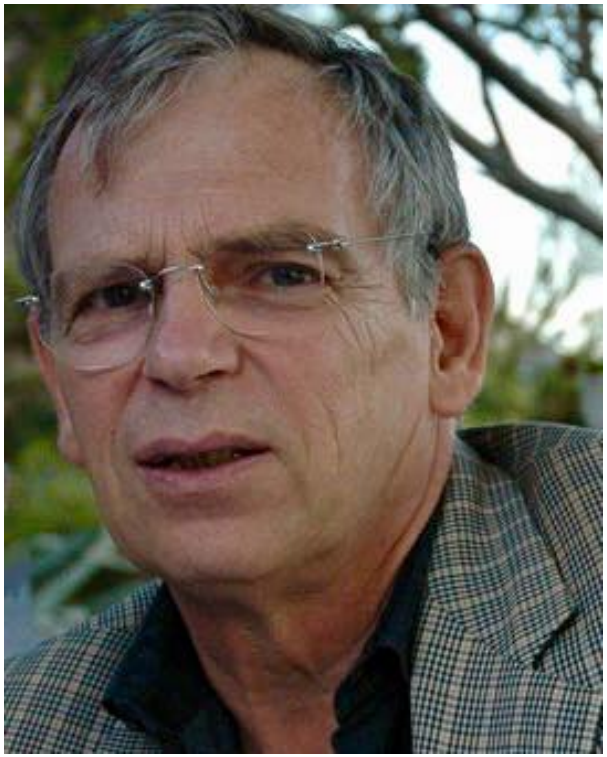
- John McCarthy1958

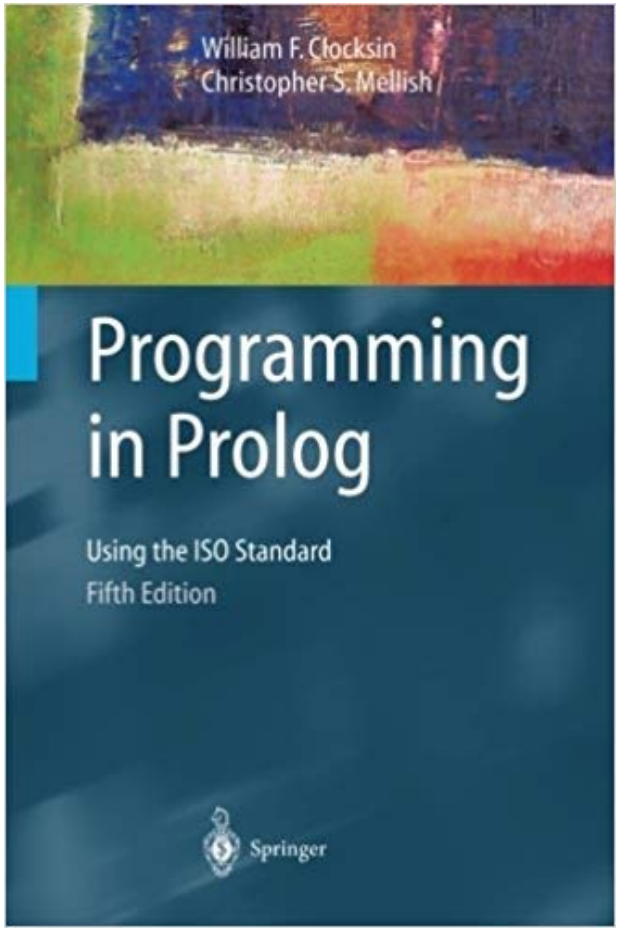
Ed Feigenbaum

The potential use of computers by people to accomplish tasks can be "one-dimensionalized" into a spectrum representing the nature of the instruction that must be given the computer to do its job. Call it the ***what-to-how spectrum****. At one extreme of the spectrum, the user supplies his intelligence to instruct the machine with precision exactly how to do his job step-by-step. ... At the other end of the spectrum is the user with his real problem. ... He aspires to communicate what he wants done ... without having to lay out in detail all necessary subgoals for adequate performance.*

- Ed Feigenbaum 1974

Alain Colmerauer and Bob Kowalski





This course

Types of Logic Programming

Types of Logic Programming:

- Database Programming
- Classical Logic Programming
- Dynamic Logic Programming
- Constraint Systems
- Answer Set Programming
- Inductive Logic Programming (i.e. Progol)

Languages:

- Datalog
- Prolog
- Epilog
- Golog
- Progol

Schedule

Mar	29	Introduction
	31	Datasets
Apr	5	Queries
	7	Examples
	12	Query Evaluation
	14	Query Optimization
	19	View Definitions
	21	Query Optimization
	26	Simple Examples
	28	Lists, Sets, Trees
May	3	Action Definitions
	5	Dynamic Systems
	10	Database Management
	12	Worksheets
	17	Applications
	19	Applications
	24	Project Reports
	26	Project Reports
	31	Project Reports

Mathematical Background

Sets

$$\{a, b, c\} \cup \{b, c, d\} = \{a, b, c, d\}$$

$$a \in \{a, b, c\}$$

$$\{a, b, c\} \subseteq \{a, b, c, d\}$$

Functions and Relations

$$f(a, b) = c$$

$$r(a, b, c)$$

Programming Background

CS 106 or equivalent

Teams

Composition

3 people each (2 or 4 okay with *good* reason)

Names:

Pansy Division
The Pumamen
Team Camembert
Mighty Bourgeoisie
Greedy Bastards
Red Hot Chili Peppers
/*v*\
X Æ A-12
Michael Genesereth

Grades

Numerical Score

10% for each of Assignments 1, 2, 3, 4,5

50% for the Term Project

Reported Grade

Based on numerical score (see above)

No curve - independent of number of students

Satisfactory = 70% and above

Extra Credit

Added to score before determining Reported Grade

Discretionary

Textbook

Series ISSN: 1939-4608

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Introduction to Logic Programming

Michael Genesereth, *Stanford University*
Vinay K. Chaudhri, *Stanford University*

"This is a book for the 21st century: presenting an elegant and innovative perspective on logic programming. Unlike other texts, it takes datasets as a fundamental notion, thereby bridging the gap between programming languages and knowledge representation languages; and it treats updates on an equal footing with datasets, leading to a sound and practical treatment of action and change." – *Bob Kowalski, Professor Emeritus, Imperial College London*

"In a world where Deep Learning and Python are the talk of the day, this book is a remarkable development. It introduces the reader to the fundamentals of traditional Logic Programming and makes clear the benefits of using the technology to create runnable specifications for complex systems." – *Son Cao Tran, Professor in Computer Science, New Mexico State University*

"Excellent introduction to the fundamentals of Logic Programming. The book is well-written and well-structured. Concepts are explained clearly and the gradually increasing complexity of exercises makes it so that one can understand easy notions quickly before moving on to more difficult ideas." – *George Younger, student, Stanford University*

ABOUT SYNTHESIS

This volume is a printed version of a work that appears in the ***Synthesis Digital Library of Engineering and Computer Science***. Synthesis books provide concise, original presentations of important research and development topics, published quickly, in digital and print formats.

MORGAN & CLAYPOOL PUBLISHERS
store.morganclaypool.com

ISBN 978-1-68173-722-5
90000
9 781681 737225

GENESERETH • CHAUDHRI
INTRODUCTION TO LOGIC PROGRAMMING
MORGAN & CLAYPOOL

MORGAN & CLAYPOOL PUBLISHERS

Introduction to Logic Programming

Michael Genesereth
Vinay K. Chaudhri

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Ronald J. Brachman, Francesca Rossi, and Peter Stone, *Series Editors*

http://cs151.stanford.edu

Introduction to Logic Programming

What versus How

The following syllabus lists all of the materials of the course. Note that there are interactive exercises at the ends of the chapters in the course textbook. (Click on the exercise numbers to go to the exercise pages.) These exercises are an essential part of the course, and you will benefit from tackling them. Some are easier than others, but you should attempt them all. Do the exercises! Do The Exercises!! DO THE EXERCISES!!!

Color Code

- Black - Lecture Slides
- Blue - Readings
- Red - Assignments
- Grey - Comment

Introduction (Week 1)

- Lecture 1 - Introduction
- Lecture 2 - Datasets
- Chapter 1 - Introduction
- Chapter 2 - Datasets
- Programs With Common Sense
- Logic Programming
- Assignment 1.1 - Datasets in Sierra
- Assignment 1.2 - Game State
- Assignment 1.3 - Triples
- Project