Logic Programming

Introduction

Michael Genesereth
Computer Science Department
Stanford University

Lecture will begin at ~2:35 PDT.
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

Inputs → Interpreter → Outputs

```
public class CreateRectAndPoint {
    public static void main(String[] args) {
        // Create a point object and two rectangle objects
        Point orig = new Point(0, 0);
        Rectangle rect1 = new Rectangle(orig, 100, 200);
        Rectangle rect2 = new Rectangle(new Point(50, 50), 150, 300);
        // Display rect1's width, height, and area
        System.out.println("Width of rect1: " + rect1.getWidth());
        System.out.println("Height of rect1: " + rect1.getHeight());
        System.out.println("Area of rect1: " + rect1.getArea());
        // Set rect1's position
        rect1.setOrigin(new Point(200, 200));
        // Display rect1's position
        System.out.println("Position of rect1: " + rect1.getOrigin());
        System.out.println("Position of rect2: " + rect2.getOrigin());
    }
}
```
A triangle is a polygon with 3 sides.

\[ e = mc^2 \]
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 basically a **runnable specification**.
A triangle is a polygon with 3 sides.
Logic as a Specification Language

Language

General purpose language
+ Highly expressive

Some special purpose languages are easier to use
Other general purpose declarative languages exist
But all can be converted to logical statements

Interpreter

Automated Reasoners capable of drawing conclusions
Can take advantage of domain-dependent reasoners
but are 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.

There is no need to make arbitrary choices.

Programs can be composed with other programs.

Easier to learn logic programming than traditional programming.

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

Ability to respond to changing circumstances
Versatility

Ability to be used for multiple purposes

Example

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
Circuit:

Premises:

\[ o \Leftrightarrow (x \land \neg y) \lor (\neg x \land y) \]
\[ a \Leftrightarrow z \land o \]
\[ b \Leftrightarrow x \land y \]
\[ s \Leftrightarrow (o \land \neg z) \lor (\neg o \land z) \]
\[ c \Leftrightarrow a \lor b \]

Applications:
Simulation
Configuration
Diagnosis
Test Generation
Deductive Databases

\[
q(X) :\neg p(X,Y) \land p(X,Z) \land Y=!Z
\]

\[
g(X,Z) :\neg p(X,Y) \land p(Y,Z)
\]

\[
illegal :\neg p(X,Y) \land p(Y,X)
\]
Data Integration

Integrated Search

Side-by-side Comparison

Integration Engine

Supplier 1
Supplier 2
Supplier 3
Supplier 4
Manufacturer 1
Manufacturer 2
Satisfaction Ratings
Product analysis
Marketplace Data
Business Rules
Worksheets

Gates Information Network

Title: Create a new Event.
Room: Gates 200
Date: 2010-10-08
Start Time: 2
End Time: 2
Duration: 2
Owner: Michael Genesereth

Comments and complaints to: action@logic.stanford.edu

DEPARTMENT OF COMPUTER SCIENCE
MSCS Program Sheet (2010-11)

Name: Charles Pamel
Primary Specialization: Artificial Intelligence

Gates Information Network

Worksheets

http://logicprogramming.stanford.edu/examples/programsheets/demonstration.html
Computational Law is that branch of legal informatics concerned with the mechanization of legal reasoning.

Automated Legal Reasoning Systems
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
Non-Successes
Natural Language Processing

Lecture Notes

PROLOG AND NATURAL LANGUAGE ANALYSIS

Fernando C.N. Pereira and Stuart M. Shieber

CENTER FOR THE STUDY OF LANGUAGE AND INFORMATION
PTTP

means

Prolog Technology Theorem Prover
Japan’s Fifth Generation Project
History
LGP-30
IBM 360

Figure 4. Card Codes and Graphics for 84 Character Set
Assembly Language

- Assembly Language
  - mov ecx, ebx
  - mov esp, edx
  - mov edx, r9d
  - mov rax, rdx

  Programmer

- Assembler + Linker

- Machine Language
  - 100101011001
  - 010011111011
  - 111010101101
  - 010101010100

  Processor
Higher Level Languages
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 McCarthy 1958
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
Bob Kowalski
This course
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
- LPS
- Progol
Schedule

Mar 30  Introduction
Apr  1  Datasets

6  Queries
8  Examples

13  Query Evaluation
15  Query Optimization

20  View Definitions
22  Query Optimization

27  Simple Examples
29  Lists, Sets, Trees

May  4  Action Definitions
6  Dynamic Systems
11  Database Management
13  Worksheets

18  General Game Playing
20  Computational Law
25  Extensions
27  Extensions

Jun  1  Project Reports
3  Project Reports
Sets

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

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

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

Functions and Relations

\quad f(a, b) = c

\quad r(a, b, c)
CS 106 or equivalent
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
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.
http://cs151.stanford.edu
Introduction to Logic Programming

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