Logic Programming

Mestrado Integrado em Engenharia Informática e Computação

Rui Camacho (LIAAD)

Henrique Cardoso (LIACC)

Daniel Silva (LIACC)

FEUP - Faculdade de Engenharia da Universidade do Porto

rcamacho@fe.up.pt , hlp@fe.up.pt e dcs@fe.up.pt

1st Semester of 2018-2019

Logic Programming Course Presentation

Objectives

- Extend the student's knowledge and experience in the paradigm of Declarative Programming
- Develop skills for abstract reasoning and of declarative problem representations
- Learn and experiment with Logic Programming
- Learn and experiment with Constraint Logic Programming
- Learn Prolog
- Learn Constraint Logic Programming

Learning outcomes

At the end of this course, students should be able to:

- Identify classes of problems where (Constraint) Logic Programming is particularly relevant
- Apply Prolog programming and constraint logic programming techniques
- Build full Prolog applications, with possible connections to other programming languages.

students should have acquired programming skills useful for other courses in the domain of Artificial Intelligence

Course Contents (I)

- Foundations of Logic Programming
  - Propositional logic
    Predicate logic
    Inference rules
    Horn clauses. Unification
    Resolution
    Conjunctive normal form
  - From logic to logic programming
    Origins of Prolog

Course Contents (II)

- Logic Programming
  - Clauses, Predicates, Facts
    Queries
    Rules
    Logic variables
    Instantiation
  - Logic programming and databases
    Recursion
    Lists, Trees
    Symbolic expressions
  - Computation model of a logic program
    The unification algorithm
    Abstract interpreter
    Traces
    Search trees
    Negation

Course Contents (III)

- The Prolog Language
  - Execution model
    Backtracking
    Rule and goal order
    Termination
  - Arithmetic
    Iteration
    Structure inspection
    Meta-logical predicates
    Control: cuts and negation
    Extra-logical predicates
- Advanced Prolog Programming Techniques
  - Non-deterministic programming
    Incomplete data structures
    Meta-interpreters
    Search techniques

Course Contents (IV)

- Constraint Logic Programming
  - Constraints
    Constraint satisfaction
    Constraints in finite domains
  - Constraint logic programming (CLP)
    Search control
    Variable and value ordering
    Modeling problems in CLP
    Constraint programming using SICStus Prolog

Bibliography

Main references:

- Sterling, Leon; The Art of Prolog. ISBN: 0-262-69163-9
- Marriot, Kim; Programming with constraints. ISBN: 0-262-13341-5
- Clocksin, W. F.; Programming in prolog. ISBN: 0-387-58350-5

Complementary references:

- Christopher Hogger, Essentials of logic programming,Oxford University Press, 1990 ISBN:0-19-853832-4
- Torres, Delfim Fernando Marado; Introdução à programação em lógica. ISBN: 972-8021-93-3
- Bratko, Ivan; Prolog programming for artificial intelligence. ISBN: 0-201-40375-7
- O.Keefe, Richard A.; The craft of Prolog. ISBN: 0-262-15039-5
- Stuart Russell, Peter Norvig; Artificial intelligence. ISBN: 978-0-13-207148-2

Software

- SICStus Prolog (http://www.sics.se/sictus/)
- Yap Prolog (http://www.dcc.fc.up.pt/∼vsc/Yap/)
- SWI Prolog (http://www.swi-prolog.org/)

Teaching Procedures

Two Types of lectures:

- Theoretical classes are used for exposition of the main (constraint) logic programming concepts, presentation and discussion of practical examples.
- Practical classes are used to solve programming exercises and for assisting students on their practical assignments.

Student's Assessment (I)

- Two intermediate "exams" made using Moodle (E1, E2)
- Two project Assignments (TP1, TP2)

Important Dates:

28th September	TP1 choice
21th October	TP1 intermediate report
18th November	TP1 final evaluation + source code
20nd November	TP2 choice
21nd November	first exam E1 (Prolog)
23rd December	TP2 report + source code
7th January 2018	second exam E2 (PLR)
??th January 2018	TP2 final evaluation

Compulsory minimal classification:

TP1 minimum mark of 7 in 20
TP2 minimum mark of 7 in 20
E1 minimum of 7 in 20
E2 minimum of 5 in 20

Student's Assessment (II)

Final Grade decomposition:

Final Grade: = 50% Exams (E1,E2) + 50% Assignments (TP1,TP2)

Exam's grades:

Exams = 65% Exam E1 + 35% Exam E2

Assignments grades:

Assignments = 65% Assignment 1 (TP1) + 35% Assignment 2 (TP2)

TP1 = 25% (Intermediate report of TP1) + 75% (final eval. of TP1)

Aplicações de Prolog

- Inteligência Artificial:
  - Jogos
  - Sistemas Baseados em Conhecimento
  - Compreensão de Linguagem Natural
- Data Mining Multi-relacional: (modelos compreensíveis)
  - Desenho racional de fármacos (Quimoinformática)
  - Aplicações em Genómica (Bioinformática)
  - Previsão da estrutura [3D] de proteínas (Quimoinformática)
  - Sistemas de *Inductive Logic Programming*
  - Previsão de efeitos adversos de medicamentos

Mesh Design

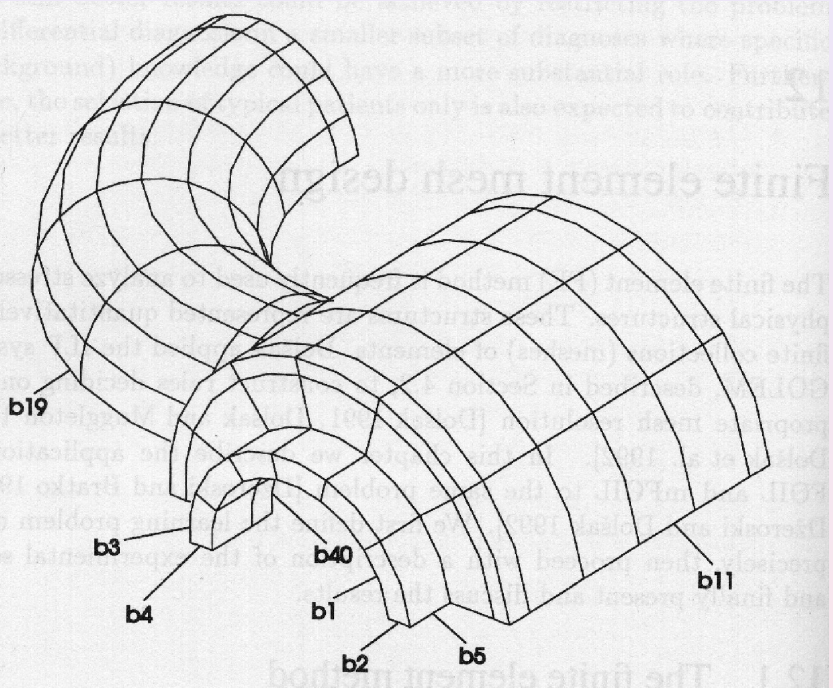
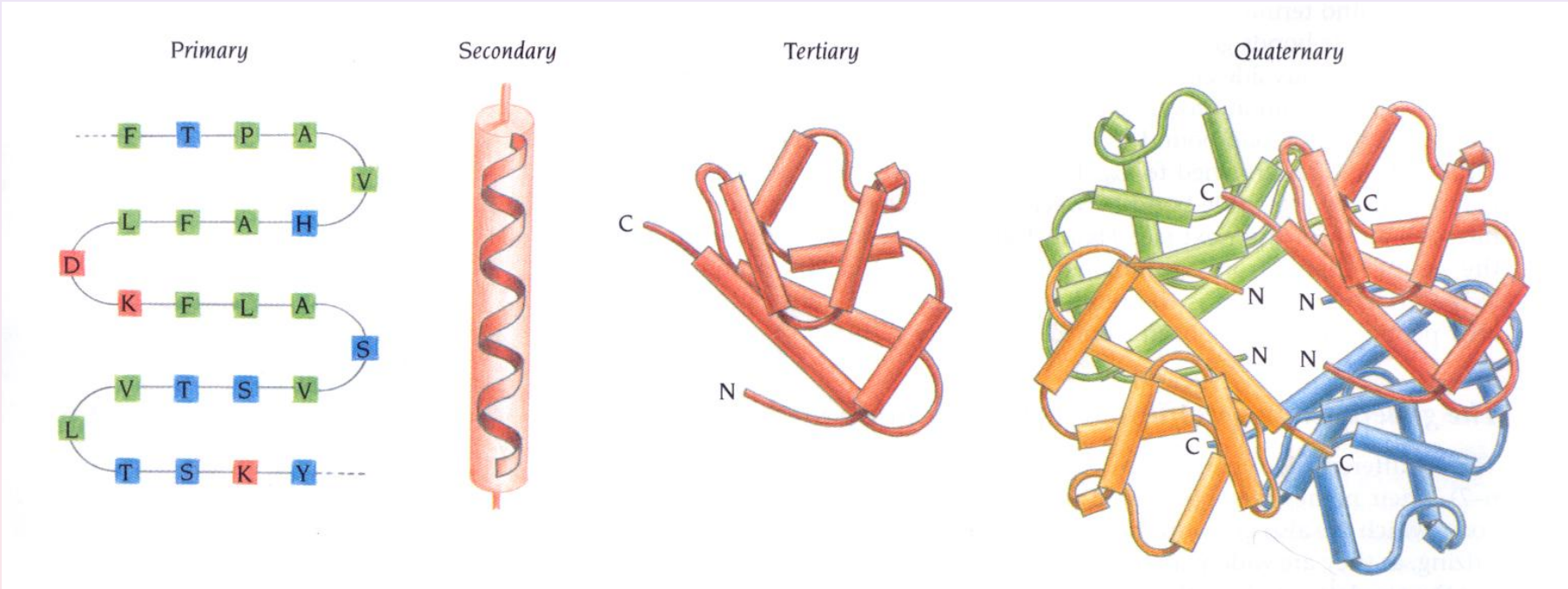
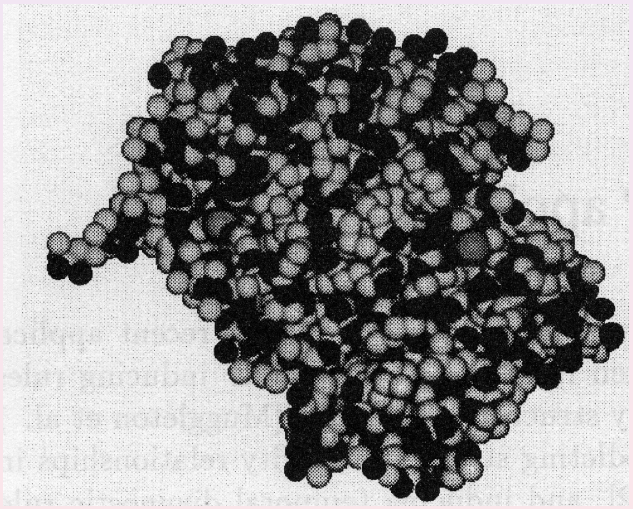


Figure 12.1: A typical structure with its corresponding FE mesh. From Dolšak [1991].

Prediction of 3-D structure of proteins



Predicting Protein 3D structure



Level 0 rule

$$alpha0(A,B) \leftarrow octf(D,E,F,G,B,H,I,J,K),\ position(A,D,Q),not\_p(Q),not\_k(Q),\ position(A,E,O),not\_aromatic(O),\ small\_or\_polar(O),\ position(A,F,R),\ position(A,G,P),not\_aromatic(P),\ position(A,B,C),very\_hydrophobic(C),\ not\_aromatic(C),\ position(A,H,M),large(M),not\_aromatic(M),\ position(A,I,L),hydrophobic(L),\ position(A,K,N),large(N),ltv(N,R).$$

Level 1 rule

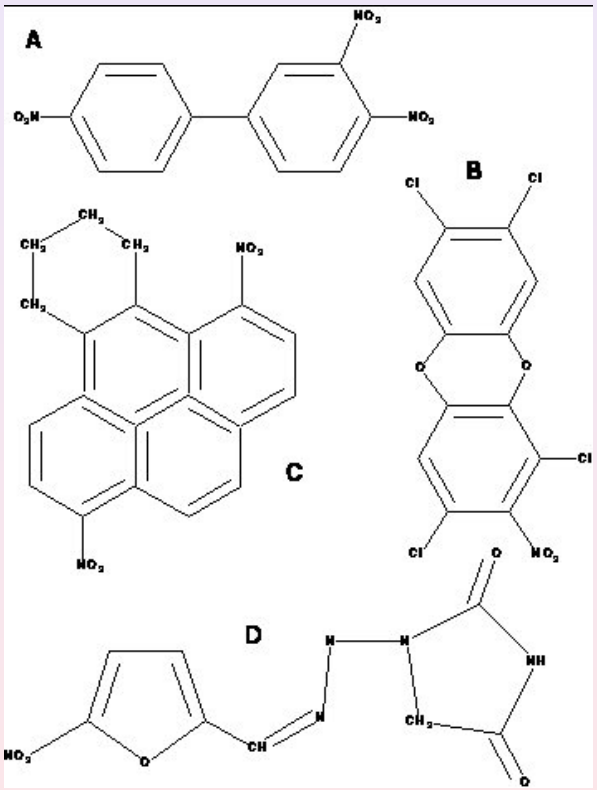
$$alpha1(A,B) \leftarrow octf(D,E,F,G,B,H,I,J,K),\ alpha0(A,F),alpha0(A,G).$$

Level 2 rule

$$alpha2(A,B) \leftarrow octf(C,D,E,F,B,G,H,I,J),\ alpha1(A,B),alpha1(A,G),alpha1(A,H).$$

Figure 14.2: Rules for predicting α-helix secondary structure.

Structure-Activity Relationship (SAR) Problems
Mutagenesis study

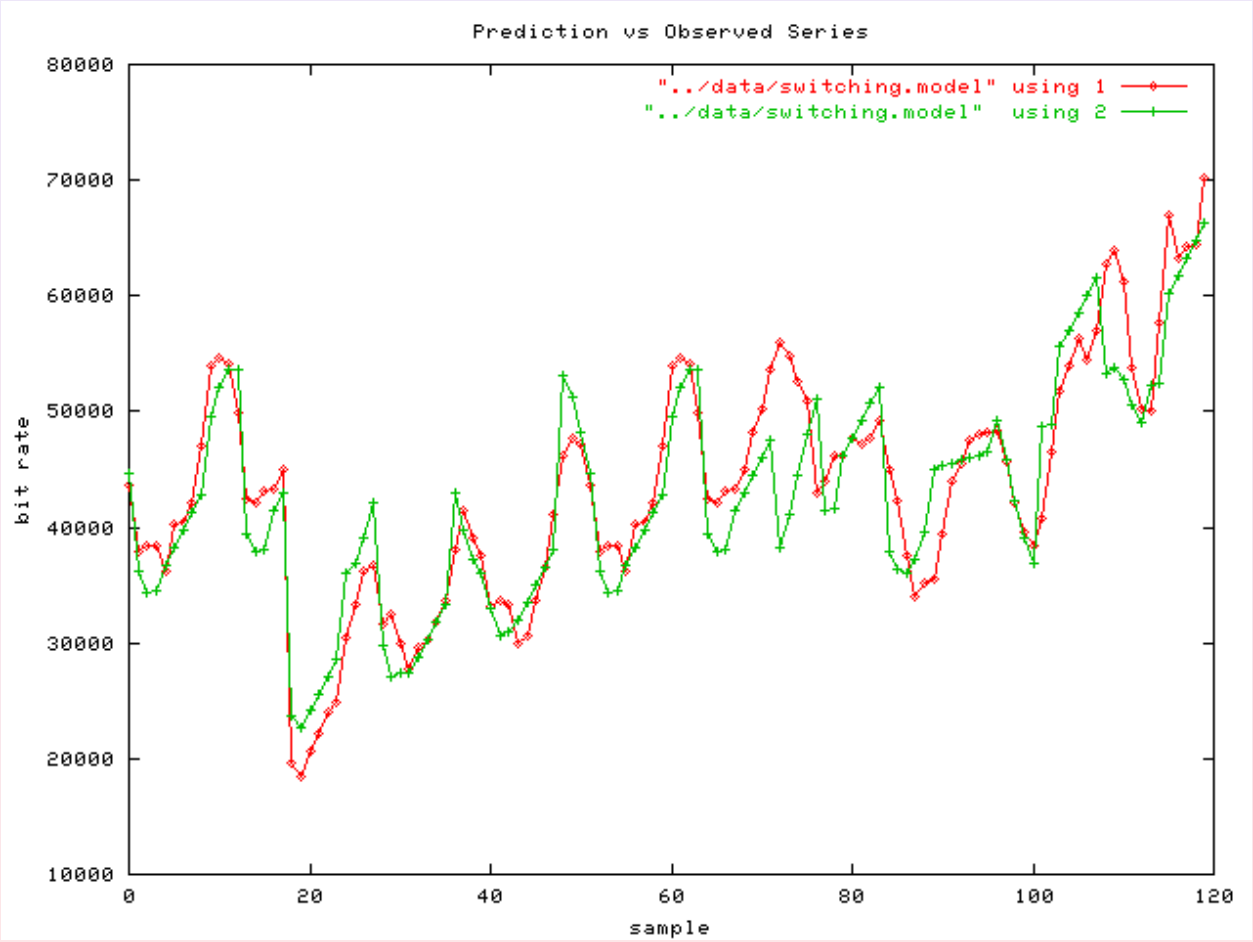


Some molecules discovered in the study:

- (A) 3,4,4'-tri-nitro-biphenyl
- (B) 2-nitro-1,3,7,8-tetrachlorodibenzo-1,4-dioxin
- (C) 1,6,-dinitro-9,10,11,12-tetrahydrobenzo[e]pyrene
- (D) nitrofurantoin

Time Series applications

Data Network Traffic Engineering (Master thesis)



Behavioural cloning – an application