CIS 730 Artificial Intelligence
CIS 530 Introduction to Artificial Intelligence
Fall 2018

Homework 4 of 10: Problem Set
Forward and Backward Chaining

Assigned: Mon 17 Sep 2018
Due: Mon 24 Sep 2018 (before midnight)

The purpose of this assignment is to develop your basic understanding of **first-order logic** and **forward and backward chaining** through problem solving by hand.

This homework assignment is worth a total of 100%.
Each problem is worth 25% for CIS 730 students and 34% for CIS 530 students.

As of Sun 16 Sep 2018, CLIPS (v6.24, 15 Jun 2006) is installed on the class application server, `fingolfin.kdd.cs.ksu.edu`, and it will also be installed on the class DigitalOcean droplet, `finarfin`. You should be able to request an account on `cislinux` if you do not already have one (see https://support.cis.ksu.edu/CISDocs/wiki/Accounts), and you will receive e-mail with information on your `fingolfin` account. For connection instructions for both systems, see: http://bit.ly/kstate-cs-linux.

- CLIPS is available for Windows, MacOS, and Linux from: http://clipsrules.sourceforge.net/
- SWI-Prolog is available for Windows, MacOS, and Linux from: http://www.swi-prolog.org/download/stable (follow the APT instructions to install builds for Ubuntu and Debian distributions: http://www.swi-prolog.org/build/Debian.html)

References

For Problems 1-2, refer to Lectures 8 and 9 and the examples from:

1. Milos Hauskrecht of the University of Pittsburgh: http://people.cs.pitt.edu/~milos/
2. Roman Barták of Charles University in Prague: https://ktiml.mff.cuni.cz/~bartak/.

For Problems 3-4, refer to:

1. The CLIPS textbook, *Expert Systems: Principles and Programming, 3rd edition* (1998) by Joseph Giarratano and Gary Riley.
2. The CLIPS 6.2.4 documentation (especially section 4 on `deffacts` and 5 on `defrule`): http://clipsrules.sourceforge.net/documentation/v624/bpg.htm
3. *Structure and Interpretation of Computer Programs*, 2e by Abelson, Sussman, and Sussman (*aka* SICP2): https://sarabander.github.io/sicp/html/3_002e3.xhtml
4. "Backward and Forward Chaining" by the award-winning math teacher Eddie Woo of Cherrybrook Technology High School, Sydney: https://youtu.be/ZhTt-GG7PiQ

Problems

1. **(530 / 730). Constraint Satisfaction Problems (CSP): 3-Coloring, Minimum Remaining Values (MRV), and Least Constraining Value (LCV).** Construct an example of a 3-coloring problem on a 10-node graph where MRV (fail-fast) and LCV (least commitment) both improve on value propagation with backtracking (*aka* "backtracking search"). Show the behavior of backtracking search by writing down a problem-solving trace. Turn in your solution as a separate file (`ps4_1.pdf`) or part of a single `ps4.pdf`.

2. **(530/730) CSP and Arc Consistency.** Now compare your solution for PS4-1 to forward checking with constraint propagation (FC/CP), and arc consistency (AC). Your solution should discuss MRV and LCV savings for FC/CP and AC-3 as defined in R&N 3^e and in class. For your example, do you get as much savings with MRV and LCV using FC/CP? Using AC-3? Why do you think this is?

   Turn in your solution to the discussion questions, `ps4_2.pdf`. This can be part of a single `ps4.pdf`.

3. **(530/730) Forward Chaining, Backward Chaining, and Production Systems.** See *Expert Systems: Principles and Programming, 3rd edition* by Joseph C. Giarratano and Gary D. Riley (see the Files/Handouts/Supplements directory). Write a simple program using CLIPS to demonstrate forward and backward chaining:

   First, write a CLIPS program that will propagate constraints for the Fahrenheit-to-Celsius converter in SICP2, Figure 3.28:
   https://sarabander.github.io/sicp/html/3_002e3.xhtml#g_t3_002e3_002e5
   This should take the form of a rule whose left-hand side (LHS) is the C value and whose right-hand side is the calculated F value.

   Second, look at the `adder` and `process-new-value` functions in §3.3.5 of SICP2. This Scheme function implements a rudimentary form of CSP solver that triggers when two of three values are given. Discuss how this naturally follows a forward chaining (FC) inference pattern and how to adapt this paradigm to FC systems such as production systems (you don't actually have to write the CLIPS rules). What is the challenge of solving this problem using a backward chaining system?

   Turn in your source (`ps4_3.clp`), a problem solving trace (`mp4_3-out.txt`), and your solution to the discussion questions, `ps4_3.pdf`. This can be part of a single `ps4.pdf`, but you must turn in the source code and output files so that your solution can be tested.

4. **(730) Implementing Forward Checking and Arc Consistency in CSP using Production Systems.** Continuing from PS4-3: do production systems confer any advantage in implementing constraint propagation? How about arc consistency? Explain how this would work in a production system: how would the rules be structured so the system would stop at the right time? (For PS4-4 you may sketch a solution rather than give a complete CLIPS implementation).

   Turn in your solution to the discussion questions, `ps4_3.pdf`. This can be part of a single `ps4.pdf`.

Class Participation (required)

Submit your **revised** term project plan as specified in class the week of Mon 10 Sep 2018 project assignment. Follow up during office hours and/or in Slack if you have not yet been checked off as having participated in a proposal interview.

Read And Explain Pairs

After going over your **Read And Explain Pairs** exercise with your assigned partner (to be assigned and described in Canvas), post a short paragraph summarizing *forward chaining* and *backward chaining* and a second containing any questions you may have on search to the class mailing list (**CIS530-L@listserv.ksu.edu** or **CIS730-L@listserv.ksu.edu**). That is, ask questions about any knowledge representation topic for which your understanding is unclear. This includes propositional logic, first-order logic, description logic, or theorem proving (forward and backward chaining, resolution strategies)

Midterm Exam Review

A review guide will be posted this Friday (21 Sep 2018) with examples not in PS3.

Coming Up Next

Machine Problem 5 (due Fri 05 Oct 2018) – Heuristic Search Part 2 of 2 & Production Systems. You will continue studying programming techniques for search by implementing uninformed and informed searches and running comparisons or empirical evaluation.

Problem Set 6 (due Mon 22 Oct 2018) – Knowledge Representation and Reasoning, Part I: Clausal Form (CNF) Conversion, Elicitation, Ontologies, Analogy. You will start to work with eliciting ontological and fact-based knowledge about a simple domain and encoding it in Protégé-OWL. You will also start getting some experience working with analogical reasoning.

Machine Problem 7 (due Mon 29 Oct 2018) – Knowledge Representation and Reasoning, Part II: Ontology Reasoning, Classical and Robust Planning. You will apply the Pellet ontology reasoner on the Protégé-OWL ontology you developed in MP5. You will also solve some planning problems to simulate the behavior of algorithms for classical and modern (reactive) planning. The Waikato Environment for Knowledge Analysis (WEKA). Finally, you will start to work with Hugin, TETRAD, and WEKA.

Machine Problem 8 (due Mon 05 Nov 2017) – Reasoning and Learning, Part I: FOL Converter; Intro to WEKA and `scikitlearn`. You will start to write a program to convert partially parsed first order predicate calculus (FOPC), *aka* first-order logic (FOL) sentences, into CNF, i.e., clausal form. You will practice converting partially parsed first order predicate calculus (FOPC), *aka* first-order logic (FOL) sentences, into CNF, i.e., clausal form, and run these through a Prolog interpreter. You will also get your first hands-on experience by running two machine learning software packages (the Waikato Environment for Knowledge Analysis (WEKA) and `scikitlearn`) to train Logistic Regression, Decision Tree, and Perceptron models.

Problem Set 9 (due Fri 09 Nov 2017) – Reasoning and Learning, Part II: Probabilistic Reasoning (Inference and Causality), Version Spaces, and Decision Trees. You will solve some Bayesian network reasoning and learning tasks and a few classification problems to simulate the behavior of supervised inductive learning algorithms to prepare for the final machine problem.

Machine Problem 10 (due Fri 16 Nov 2017) – Perception and Understanding: Using WEKA; Artificial Neural Networks (ANNs), Genetic and Evolutionary Computation (GEC), Natural Language Processing (NLP), and Vision. You will apply learning algorithms in WEKA and `scikitlearn`, plus an ANN or GEC, to a classification task and to a pattern recognition or obstacle avoidance task for a simple mobile robot. You will solve problems and answer some discussion questions about perception and understanding. (This will include some practice final exam questions.)