Massachusetts Institute of Technology

16.410-13 Principles of Autonomy and Decision Making

Due: Friday, 12/03/04

Paper solutions are due no later than 5pm on Friday, 12/03/04. Please give solutions to course secretary Brian O' Conaill, at his desk outside of 33-330.

Objectives

The first objective of this problem set is to develop your facility with propositional logic and the DPLL algorithm. For propositional logic, the objective is to develop your skill at encoding English statements in propositional logic, determining the truth of a sentence with respect to an interpretation, and reducing a sentence to clausal form. For satisfiability, the objective is to exercise your understanding of unit propagation and DPLL (backtrack search plus unit propagation).

The second objective is to develop your understanding of model-based diagnosis. This includes applying the concepts of constraint suspension, conflict extraction, candidate generation and candidate testing.

The third objective is to develop your understanding of decision tree learning. This includes the generation of a decision tree from a data set, and the application of the decision tree to new instances.

16.410 Assignment

For 16.410 students, this problem set is comprised of **both** an **online portion** and a **written portion**, given below. 16.410 students should log into the online tutor, and complete the online problems under problem set #9 by the due date, listed above. 16.410 students must complete the threee problems listed below, turning in hardcopy solutions as indicated above.

16.410 Assignment

16.413 students **do not** have any **online problems** assigned, **only** a **written assignment**. Please complete **the three problem**s listed below and turn in hardcopy solutions to the course secretary, as indicated above.

Problem 1 Propositional Logic and Satisfiability

Consider the following quote, taken from (Barwise and Etchemendy, 1993):

> If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a [mortal] mammal, then it is horned. The unicorn is magical if it is horned.

Represent this quote using the following propositional symbols:

MY	=	"Mythical Unicorn"
IU	=	"Immortal Unicorn"
MM	=	"Mortal Mammal"
HU	=	"Horned Unicorn"
MG	=	"Magical Unicorn"

Part a. Using the above propositional symbols, provide a propositional logic sentence S that is a **direct translation** of the above quote.

Part b. Determine the truth of your sentence S, from part a, for the following interpretation I:

{MY = False, IU = False, MM = True, HU = True, MG = True}

- Show how you derive the truth of S, using I and the semantics of the logical connectives. Your derivation should be similar to the derivation given in class.

Part c. Reduce your sentence S, from part a, to a set of propositional clauses.

- Walk through your reduction step by step, starting from sentence S and ending with your set of clauses. Your reduction should be similar in style to the reduction given in the appendix of the class notes.

- List your resulting clauses, labeling them C1 … Cn.

Part d. Add the following clause, C0, to the set of clauses C1 … Cn, from part c:

C0: $\neg$ MG

Apply unit propagation to clauses C0, C1 … Cn.

- List the truth assignment (one of "True," "False" or "Unassigned") that unit propagation determines for each proposition. For each proposition that is assigned "True" or "False," give its support, that is, the clause and assignments used to determine the proposition's truth.
- List all clauses that are violated by this truth assignment.
- List all clauses that are satisfied by this truth assignment.

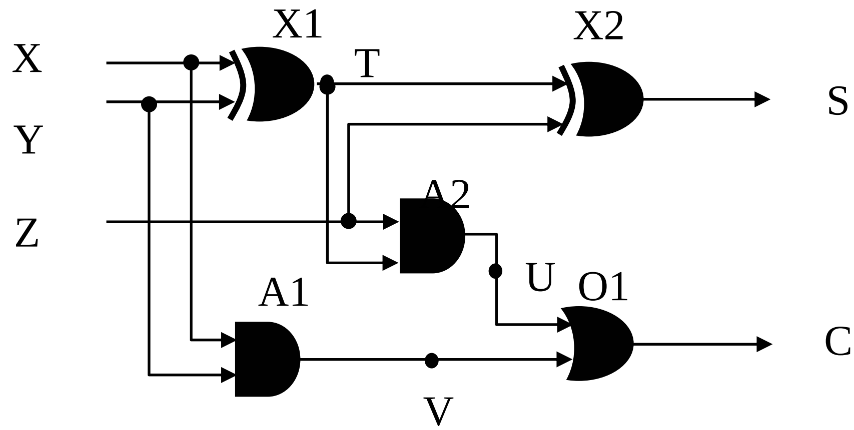
Part e. Use the DPLL algorithm (Backtrack Search + Unit Propagation) to enumerate **all models** of clauses C1 … Cn, from part c (**do not** include C0). Recall that a model = assigns True or False to all propositions, such that all clauses are satisfied.

You must assign the propositions in the order: MY, IU, MM, HU and, finally, MG. For each proposition, first assign "True," followed by "False."

- Give the **full** backtrack search tree resulting from applying DPLL, that is, run the search until **all** subtrees have been explored or pruned. Label each branch with the assignment made for that branch. In addition, next to each branch, list all assignments that are derived by unit propagation for that branch. Cross off any node that is pruned as inconsistent. Circle any (leaf) node that denotes a model.
- List all models that you find by applying DPLL.

Problem 2 – Model-based Diagnosis

Consider the application of Single_Fault_w_Conflicts (constraint suspension) to the full adder circuit, shown in the schematic below.



The full adder is a digital circuit that has three binary inputs (X, Y and Z), and computes a two bit sum (00, 01, 10, 11), where the higher order bit of the sum is the output C and the lower order bit of the sum is S. For example, if X = 1, Y = 0 and Z = 1, then the result is C = 1 and S = 0.

The circuit consists of two eXclusive OR gates (X1 and X2), two AND gates (A1 and A2) and one OR gate (O1). In constraint suspension, we model each component C as having two possible modes: the good mode, denoted G(C), and the unknown mode, denoted U(C). If a component is working correctly, then it is in the good mode; otherwise, it is in the unknown mode.

Each component has no behavior specified when in the unknown mode, that is, its constraints are "suspended." This means that any behavior is possible (consistent) in the unknown mode.

Each component has its standard behavior when in the good mode; that is:

- the output of an AND gate is 1 if and only if both its inputs are 1,
- the output of an OR gate is 0 if and only if both its inputs are 0, and
- the output of an exclusive OR gate is 1 if and only if one input is 1 and the other input is 0.

Suppose you observe the following inputs and outputs for the full adder:

OBS = { X = 0, Y = 0, Z = 1, S = 0 and C = 1}

Given OBS and the model described above, diagnose the circuit using the algorithm Single_Fault_w_Conflicts, as presented in lecture.

Part a. Candidate Generation

Generate the initial candidate set. First assume that all components are in the "good" mode G. Use propagation to derive a symptom, and extract a conflict.

- List the conflict found.
- List all candidates determined from this conflict.

Part b. Derive Diagnoses by Testing Candidates

Identify the candidates from part a that are diagnoses, by applying the algorithm Single_Fault_Test_Candidates.

- For each candidate, in the order tested, list:
  - the candidate tested,
  - each assignment to a variable that is derived by propagation,
  - whether or not the candidate is consistent with the model and observations, and
  - a conflict extracted when the candidate is inconsistent,

- List your final diagnoses.

Problem 3 Decision Tree Learning

You are using a rover on Mars to determine whether or not rocks you encounter are Martian in origin or are extra-Martian meteorites. You have collected eight samples, and used a human expert to determine whether or not the eight sample rocks are meteorites. You now want the rover to learn to classify meteorites by itself using the labeled examples.

Example	IsHeavy	IsShiny	IsSpotted	IsSmooth	IsMeteorite
A	0	0	0	0	0
B	0	0	1	0	0
C	1	1	0	1	0
D	1	0	0	1	1
E	0	1	1	0	1
F	0	0	1	1	1
G	0	0	0	1	1
H	1	1	0	0	1
U	1	1	1	1	?
V	0	1	0	1	?
W	1	1	0	0	?

You know whether or not rocks A through H are meteorites, but you do not know about U through W.

Part a. Build an ID-3 decision tree to classify rocks.

Part b. Classify rocks U, V and W as meteorites or not, using this decision tree.