Linköpings Universitet
Institutionen för Datavetenskap
Patrick Doherty

Tentamen
TDDC17 Artificial Intelligence
4 January 2017 kl. 14-18

Points:

The exam consists of exercises worth 41 points.
To pass the exam you need 21 points.

Auxiliary help items:

Hand calculators.

Directions:

You can answer the questions in English or Swedish.
Use notations and methods that have been discussed in the course.
In particular, use the definitions, notations and methods in appendices 1-3.
Make reasonable assumptions when an exercise has been under-specified.
State these assumptions explicitly in your answer.
Begin each exercise on a new page.
Write only on one side of the paper.
Write clearly and concisely.

Jourhavande: Piotr Rudol, 070 0895340. Piotr will arrive for questions around 16.00.

1. Alan Turing proposed the Turing Test as an operational definition of intelligence.

   (a) Describe the Turing Test using your own diagram and explanations. **[2p]**

   (b) Do you believe this is an adequate test for machine intelligence? Justify your answer. **[1p]**

2. Consider the following logical theory about registered voters (where $\mathsf{hilary}, \mathsf{bernie}$ and donald are constants) and we view grounded atomic formulas as propositional atoms. (In this case unification of two grounded atomic formulas is successful when they are identical.):

$$Democrat(\mathsf{hilary}) \tag{1}$$

$$Dislikes(\mathsf{hilary}, \mathsf{bernie}) \tag{2}$$

$$Dislikes(\mathsf{bernie}, \mathsf{donald}) \tag{3}$$

$$(Dislikes(\mathsf{hilary}, \mathsf{bernie}) \wedge Dislikes(\mathsf{bernie}, \mathsf{donald})) \rightarrow Dislikes(\mathsf{hilary}, \mathsf{donald}) \tag{4}$$

$$\neg(\neg Republican(\mathsf{donald}) \wedge \neg Democrat(\mathsf{donald})) \tag{5}$$

$$\neg Democrat(\mathsf{donald}) \tag{6}$$

   We would like to show using resolution that a registered Democrat dislikes a registered Republican. To do this, answer the following questions:

   (a) Convert formulas (1) - (6) into conjunctive normal form (CNF) with the help of appendix 1. **[1p]**

   (b) Prove that $(Democrat(\mathsf{hilary}) \wedge Republican(\mathsf{donald}) \wedge Dislikes(\mathsf{hilary}, \mathsf{donald}))$ is a logical consequence of (1) - (6) using the resolution proof procedure. **[3p]**

   - Your answer should be structured using one or more resolution refutation trees (as used in the book or course slides).

3. Based on the material in Chapter 2 of the course book, define succinctly in your own words, the following terms:

   (a) Agent, Agent Function, Agent Program **[1p]**

   (b) Performance Measure, Rationality, Autonomy **[1p]**

   (c) Reflex Agent. Also provide a schematic diagram of such an agent. **[1p]**

   (d) Model-based Agent. Also provide a schematic diagram of such an agent. **[1p]**

   (e) Goal-based Agent. Also provide a schematic diagram of such an agent. **[1p]**

4. Constraint satisfaction (CS) problems consist of a set of variables, a value domain for each variable and a set of constraints. A solution to a CS problem is a consistent set of bindings to the variables that satisfy the constraints.

   (a) Suppose there are 5 territories T1, T2, T3, T4, and T5, each with a sensor that monitors the area associated with that territory. Each sensor has three possible radio frequencies, F1,F2, F3. Sensors overlap if they are in adjacent areas. The adjacency relation between two territories is symmetric. Let $Adj(x, y)$ represent the adjacency relation where $Adj(T1, T2)$, $Adj(T1, T3)$,$Adj(T2, T3)$, $Adj(T3, T4)$. If two sensors overlap, they can not use the same frequency.

      1. Define a constraint satisfaction problem for this scenario. **[1p]**
      2. Provide a constraint graph for the CS problem. **[1p]**
      3. Provide one solution for the CS problem. **[1p]**

5. The following questions pertain to automated planning.

   (a) Explain the main ideas underlying landmark heuristics. **[2p]**

      i. For example, what is a fact landmark for a state s in a planning problem P? That is, what type of entity is it that may or may not qualify as a fact landmark, and what are the requirements for this entity to actually be a fact landmark for s given the problem P?

      ii. Also, explain at least one way in which landmarks can be used to define a heuristic function $h(s)$ in a planning problem P. Make sure to clearly specify how the *value* of the heuristic function would be calculated given the state and problem.

   (b) The following pertains to partial-order planning. **[3p]**

      i. In partial-order planning, what is a *flaw*? Name two distinct types of flaw and provide a clear explanation of each type. These explanations should be sufficiently clear for someone to understand how a partial plan can be analyzed and all flaws can be extracted.

      ii. For each type of flaw, explain at least one way of resolving it.

6. A* search is the most widely-known form of best-first search. The following questions pertain to A* search:

   (a) Explain what an *admissible* heuristic function is using the notation and descriptions in (c). **[1p]**

   (b) In the course book, we considered the 8-puzzle. This consists of a $3x3$ game board with 8 tiles (and one empty space) in the 9 slots. Each tile is numbered from 1 to 8. The start state of the game places the 8 numbered tiles arbitrarily on the board. The goal state is for the tiles to be numbered in numerical order from top to bottom, left to right. There is one action in the game. A tile can be moved from slot A to slot B if slot B is empty.

      i. The Hamming distance measures the number of misplaced tiles in a board configuration. Is this an admissible heuristic? Explain why this is the case. **[1p]**

      ii. Provide an additional admissible heuristic for this problem and explain why it is an admissible heuristic. **[1p]**

      iii. Compare your heuristic with the Hamming distance heuristic and state what heuristic dominates the other. Why is the more dominant heuristic better? **[2p]**

   (c) Let $h(n)$ be the estimated cost of the cheapest path from a node n to the goal. Let $g(n)$ be the path cost from the start node n_0 to n. Let $f(n) = g(n) + h(n)$ be the estimated cost of the cheapest solution through n.

   Provide a sufficiently rigid proof that A* is optimal if $h(n)$ is admissible. You need only provide a proof for either tree-search (seminar slides) or graph-search (in course book). If possible, use a diagram to structure the contents of the proof to make it more readable. **[2p]**

7. The following questions pertain to machine learning. Give detailed answers.

   (a) Explain how a loss function is used during training in supervised learning. **[1p]**

   (b) Describe an exploration strategy in the context of reinforcement learning. **[1p]**

   (c) Consider a Q-learning agent with the update equation shown below.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R(s_t) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \tag{7}$$

      i. How would the overall behavior of the agent change if the value of γ was reduced? **[1p]**

      ii. What is the purpose of α? **[1p]**

8. The following questions pertain to Answer Set Programming. Appendix 3 may be useful to use:

   (a) Given the program Π_1, consisting of the following rules (where r1 is a constant):

   r1: $rocket(\mathsf{r1})$.
   r2: $launches(\mathsf{r1}) \leftarrow rocket(\mathsf{r1}), not\ ab(\mathsf{r1})$.
   r3: $defect(\mathsf{r1})$.
   r4: $ab(\mathsf{r1}) \leftarrow defect(\mathsf{r1})$.

   1 what is the reduct Π_1^S for Π_1 given that $S = \{rocket(\mathsf{r1}), ab(\mathsf{r1})\}$? **[1p]**
   2 what is the reduct Π_1^S for Π_1 given that $S = \{rocket(\mathsf{r1}), defect(\mathsf{r1})\}$? **[1p]**

   (b) Given the program Π_2 consisting of the following two rules (where republican, democrat are constants):

   r1: $color(\mathsf{blue}) \leftarrow not\ color(\mathsf{red})$
   r2: $color(\mathsf{red}) \leftarrow not\ color(\mathsf{blue})$.

   What are the *possible* answer sets for Π_2? **[1p]**

   (c) Given the *possible* answer sets for Π_2 above, which of those possible answer sets are in fact answer sets for program Π_2? **[2p]**

   When answering this question be sure to show why, by using the reducts, $\Pi_2^{S_i}$, where S_i is instantiated to each possible answer set, and the consequence operator T_Π described in appendix 3.

   (d) Why is Answer Set Programming considered to be a nonmonotonic reasoning formalism? **[1p]**

9. Use the Bayesian network in Figure 1 below together with the associated conditional probability tables to answer the following questions. Appendix 2 may be helpful to use. If you do not have a hand-held calculator with you, make sure you set up the solution to the problems appropriately for partial credit.

   The conditional probability tables (CPT) should be interpreted as follows: For each row in a CPT, the left-hand side enumerates the values of the evidence variables, and each column in the right-hand side provides the probability of the query variable for each possible value of the query variable. For instance, for the CPT associated with the random variable SP, the first row denotes that $P(SP = \mathsf{High} \mid SM = \mathsf{Good}, OI = \mathsf{Good},) = 0.80$ and $P(SP = \mathsf{Low}, \mid SM = \mathsf{Good}, , OI = \mathsf{Good},) = 0.20$.

   (a) Given the Bayesian network below, write down the full joint distribution it represents. In other words, define $P(SP, SM, OI, IR)$ as a product of the conditional relationships that can be derived from the network below. **[1p]**

   (b) What is $P(SP = \mathsf{High}, SM = \mathsf{Bad}, OI = \mathsf{Bad}, IR = \mathsf{High})$? **[1p]**

   (c) What is $P(SP = \mathsf{Low} \mid SM = \mathsf{Good}, IR = \mathsf{High})$? **[2p]**

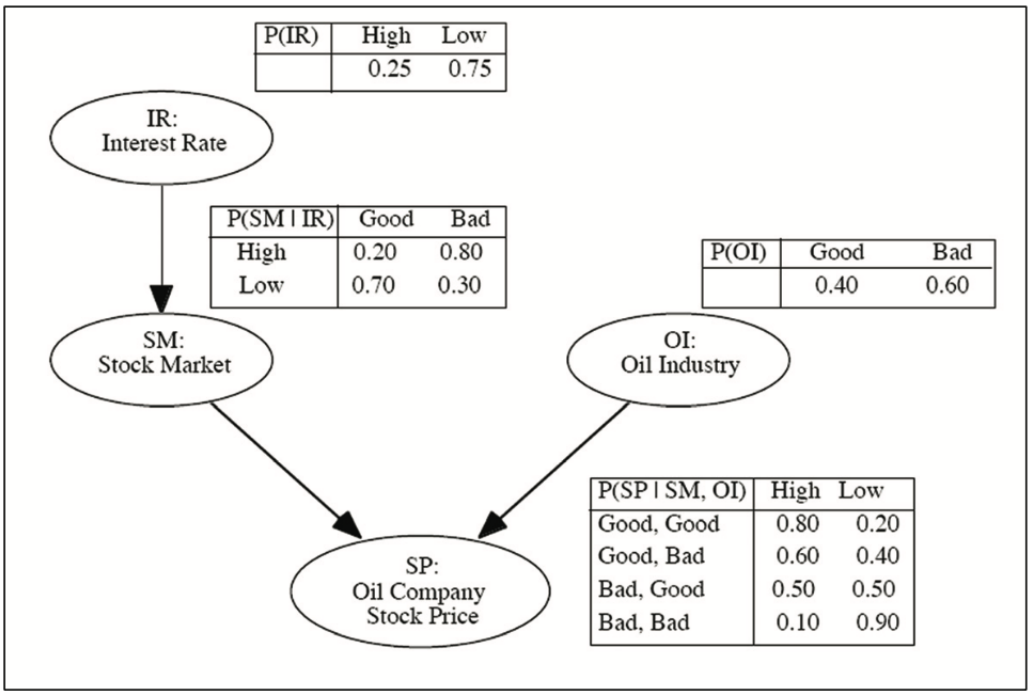

Figure 1: Bayesian Network Example

Appendix 1

Converting arbitrary wffs to CNF form: (Propositional/grounded 1st-order formula case)

1. Eliminate implication signs using the equivalence: $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$.
2. Reduce scopes of negation signs using De Morgan's Laws:
   - $\neg(\omega_1 \vee \omega_2) \equiv \neg\omega_1 \wedge \neg\omega_2$
   - $\neg(\omega_1 \wedge \omega_2) \equiv \neg\omega_1 \vee \neg\omega_2$
3. Remove double negations using the equivalence: $\neg\neg\alpha \equiv \alpha$.
4. Put the remaining formula into conjunctive normal form. Two useful rules are:
   - $\omega_1 \vee (\omega_2 \wedge \omega_3) \equiv (\omega_1 \vee \omega_2) \wedge (\omega_1 \vee \omega_3)$
   - $\omega_1 \wedge (\omega_2 \vee \omega_3) \equiv (\omega_1 \wedge \omega_2) \vee (\omega_1 \wedge \omega_3)$
5. Eliminate $\wedge$ symbols so only clauses remain.

Appendix 2

A generic entry in a joint probability distribution is the probability of a conjunction of particular assignments to each variable, such as $P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$. The notation $P(x_1, \ldots, x_n)$ can be used as an abbreviation for this.

The chain rule states that any entry in the full joint distribution can be represented as a product of conditional probabilities:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1) \tag{8}$$

Given the independence assumptions implicit in a Bayesian network a more efficient representation of entries in the full joint distribution may be defined as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \mathit{parents}(X_i)), \tag{9}$$

where $\mathit{parents}(X_i)$ denotes the specific values of the variables in $\mathit{Parents}(X_i)$.

Recall the following definition of a conditional probability:

$$\mathbf{P}(X \mid Y) = \frac{\mathbf{P}(X \wedge Y)}{\mathbf{P}(Y)} \tag{10}$$

The following is a useful general inference procedure:

Let X be the query variable, let $\mathbf{E}$ be the set of evidence variables, let $\mathbf{e}$ be the observed values for them, let $\mathbf{Y}$ be the remaining unobserved variables and let α be the normalization constant:

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y}) \tag{11}$$

where the summation is over all possible $\mathbf{y}$'s (i.e. all possible combinations of values of the unobserved variables $\mathbf{Y}$).

Equivalently, without the normalization constant:

$$\mathbf{P}(X \mid \mathbf{e}) = \frac{\mathbf{P}(X, \mathbf{e})}{\mathbf{P}(\mathbf{e})} = \frac{\sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})}{\sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbf{P}(\mathbf{x}, \mathbf{e}, \mathbf{y})} \tag{12}$$

Appendix 3: Answer Set Programming

Computing Answer Sets for a program Π:
Given a program Π:

1. Compute the possible answer sets for Π:
   (a) Powerset 2^{Π} of all atoms in the heads of rules in Π.
2. For each $S \in 2^{\Pi}$:
   (a) Compute the reduct Π^S of Π.
   (b) If $Cn(\Pi^S) = S$ then S is an answer set for Π.
   (c) If $Cn(\Pi^S) \neq S$ then S is not an answer set for Π.

The following definitions may be useful:

Definition 1 A program Π consists of a signature Σ and a collection of rules of the form:

$$l_0 \vee, \ldots, \vee l_i \leftarrow l_{i+1}, \ldots, l_m, not\ l_{m+1}, \ldots, not\ l_n$$

where the l's are literals in Σ. □

Definition 2 [Satisfiability]
A set of (ground) literals satisfies:

1. l **if** $l \in S$;
2. *not* l **if** $l \notin S$;
3. $l_1 \vee \ldots l_n$ **if** *for some* $1 \leq i \leq n, l_i \in S$;
4. a set of (ground) extended literals **if** S satisfies every element of this set;
5. rule r **if** , whenever S satisfies r's body, it satisfies r's head. □

Definition 3 [Answer Sets, Part I]
Let Π be a program not containing default negation (i.e., consisting of rules of the form):

$$l_0 \vee, \ldots, \vee l_i \leftarrow l_{i+1}, \ldots, l_m.$$

An *answer set* of Π is a consistent set S of (ground) literals such that

1. S satisfies the rules of Π and
2. S is minimal (i.e., there is no proper subset of S that satisfies the rules of Π. □

Appendix 3 is continued on the next page.

Definition 4 [Answer Sets, Part II]
Let Π be an arbitrary program and S be a set of ground literals. By Π^S we denote the program obtained from Π by

1. removing all rules containing *not* l such that $l \in S$;
2. removing all other premises of the remaining rules containing *not*.

S is an answer set of Π if S is an answer set of Π^S.
We refer to Π^S as the *reduct* of Π with respect to S. □

Definition 5 [Consequence operator T_Π]
The smallest model, $Cn(\Pi)$, of a positive program Π can be computed via its associated *consequence operator* T_Π. For a set of atoms X we define,

$$T_\Pi X = \{head(r) \mid r \in \Pi \textit{ and } body(r) \subseteq X\}.$$

Iterated applications of T_Π are written as T_Π^j for $j \geq 0$, where

$$T_\Pi^0 X = X$$
$$T_\Pi^i X = T_\Pi T_\Pi^{i-1} X \textit{ for } i \geq 1.$$

For any positive program Π, we have $Cn(\Pi) = \bigcup_{i \geq 0} T_\Pi^i \emptyset$.
Since T_Π is monotonic, $Cn(\Pi)$ is the smallest fixpoint of T_Π. □