NTNU

Norwegian University of
Science and Technology

Department of Computer and Information Science

Examination paper for TDT 4171 Artificial Intelligence Methods

Academic contact during examination: Odd Erik Gundersen

Phone: 47 63 70 75

Examination date: May 24th 2016

Examination time (from-to): 09.00 – 13.00

Permitted examination support material: No printed or hand-written support material is allowed. A specific basic calculator is allowed.

Other information: In case of uncertainty or any amiguities, the English version of the exam will be used as the reference.

Language: English

Number of pages (front page excluded): 3

Number of pages enclosed:

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig ☐ **2-sidig** ☐

sort/hvit ☐ **farger** ☐

Checked by:

Date Signature

Task 1: Reasoning with Uncertainty (20p)

a) **Independence (4p):** Explain independence and conditional independence using your own words and mathematical formulas.

Independence (2p): Two variables are independent if they do not influence each other. That is knowing about the probability of one does not not change the probability of the other. For example, the probability of a dice throw is not dependent on the outcome of past dice throws.

$$\mathbf{P}(X|Y) = \mathbf{P}(X) \text{ and } \mathbf{P}(Y|X)= \mathbf{P}(Y) \text{ and } \mathbf{P}(X,Y)=\mathbf{P}(X)\ \mathbf{P}(Y)$$

Conditional independence (2p): Two variables, X and Y, are conditional independent given a third variable, Z, if given knowledge of Z occurs, knowledge of X occurs provides no likelihood of Y occurring, and knowledge of whether Y occurs provide no likelihood of X occuring. For example, a probe catching a cavity is independent from toothache, but both toothache and the catch are dependent on whether the tooth has a cavity or not.

$$\mathbf{P}(X,Y|Z) = \mathbf{P}(X|Z)\ \mathbf{P}(Y|Z)$$

b) **Complexity (4p):** Explain why independence and conditional independence is useful when reasoning with uncertainty.

Independence and conditional independence are useful when reasoning with uncertainty because they reduce the complexity of the inference and the representation of the domain. Both independence and conditional independence enable the full joint probability tables to be reduced from exponential growth to linear growth, albeith full independene reduce the growth more.
Independence allows complete set of variables to be divided into independent subsets and then the full joint distribution can be factored into separate joint distributions on those subsets. For example, the full joint distribution on the outcome of *n* independent coin flips, $\mathbf{P}(C_1, \ldots, C_n)$ has 2^n entries, but it can be represented as the procuct of *n* single-variable distributions $\mathbf{P}(C_i)$. Hence, when independence assertions are available, they can help in reducing the size of the domain representation and the complexity of the inference problem.
Conditional independence also allows the full joint distribution to be divided into smaller tables that contain the conditional probability distributions. Because the probabilities sum to one the conditional probabilities can be further reduced, and instead of an exponential growth, $O(2^n)$, in probabilities, linear growth, $O(n2^k)$, can be achieved where k is the maximum number of possible values a variable can have.

c) **Inference (4p):** Given the full joint probability distribution (table 1), calculate P(Heart Disease | Chest Pain), first with and then without normalization.

Table 1: Heart disease.

	Chest Pain		¬Chest Pain	
	High Blood Pressure	¬High Blood Pressure	High Blood Pressure	¬High Blood Pressure
Heart Disease	0.09	0.05	0.07	0.01
¬Heart Disease	0.02	0.08	0.03	0.65

P(heartdisease | chestpain) = P(heartdisease, chestpain) / P(chestpain) =

(0.09+0.05)/(0.09+0.05+0.02+0.08) = 0.14/0.24 = 0.58

P(Heart Disease | chestpain) = α P(Heart Disease,chestpain) =

α [P(Heart Disease,chestpain,highbloodpressure)+ P(Heart Disease, chestpain, ¬highbloodpressure)] =

α [<0.09,0.02>+<0.05,0.08>] = α <0.14,0.10> = <0.58,0.42>

The task was not clear with regards to what should be distributions and not. 3p if one of the equations above is correct, and 4p if both are.

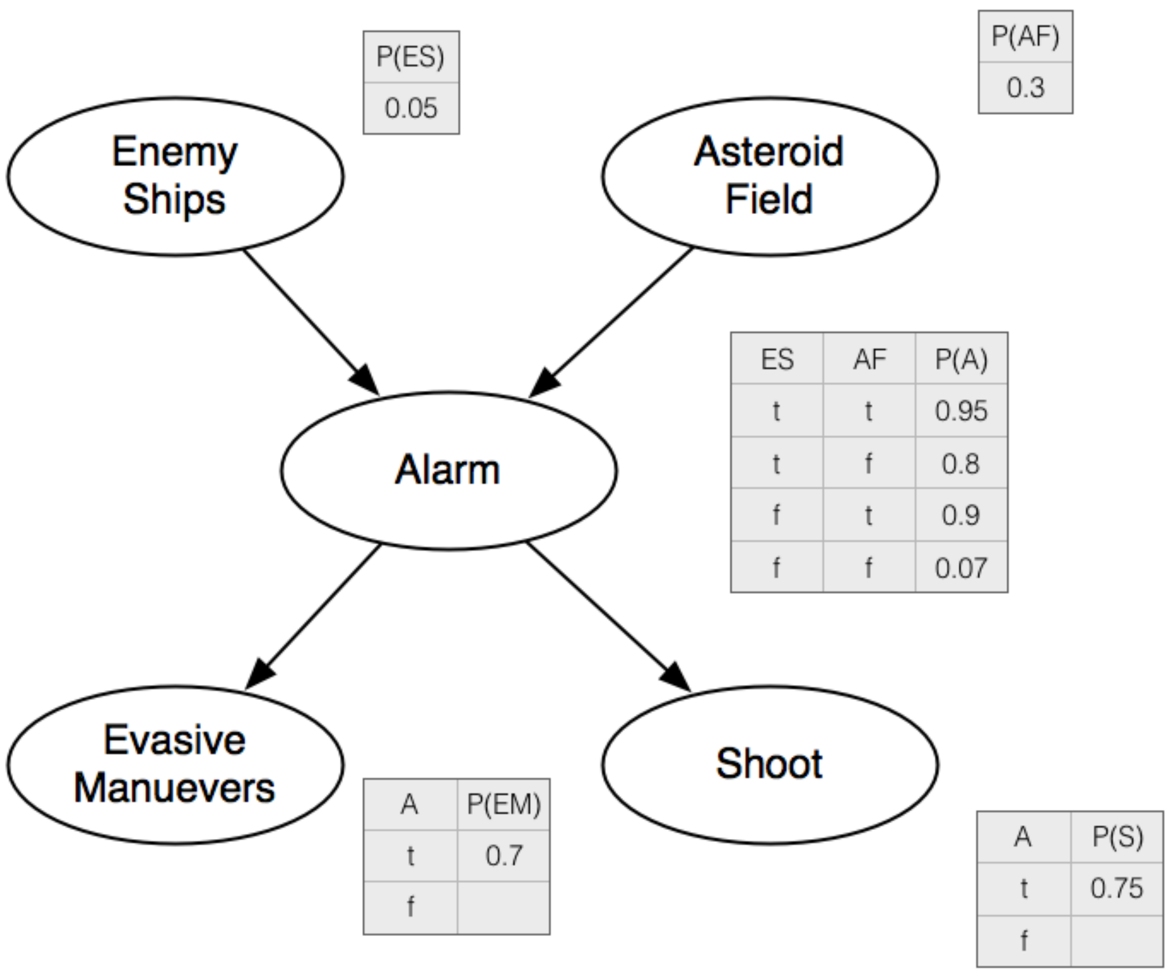
d) **Bayesian Network (8p):** Draw a Bayesian Network and set up conditional probability tables based on the following information.

The captain of the space ship the *Centurion Sparrow* starts evasive maneuvers once the alarm goes off with a probability of 0.7, while the probability that the gunner starts shooting is 0.75. The alarm goes off when the Centurion Sparrow flies into an asteroid field with a probability of 0.9. The alarm also goes off when enemy space ships are close by, but only with a probability of 0.8. If an enemy space ship appears when the Centurion Sparrow is in an asteroid field, the alarm goes off with a probability of 0.95. Sometimes, the alarm goes off when no enemy ships are close and the space ship is in open space.

This happens with a probability of 0.07. The captain gets excited when flying in asteroid fields, and she does it as often as she can. Thus, the probability of the ship being in an asteroid field is 0.3. Enemy spaceships appear with a probability of 0.05.

Now, calculate the probability that the gunner starts shooting, when ~~no~~[1] *the* alarm sounds, the space ship is in open space, the captain does nothing and an enemy ship appears.

Diagram: 3p, CPT: 3p, probability calculation: 2p



P(ES)
0.05

P(AF)
0.3

ES	AF	P(A)
t	t	0.95
t	f	0.8
f	t	0.9
f	f	0.07

A	P(EM)
t	0.7
f	

A	P(S)
t	0.75
f	

The idea was to have students have to interprete and make sense out of an informal text. This kind of backfired as the text can be interpreted in two different ways, P(S,A,¬AF,¬EM,ES) or P(S|A,¬AF,¬EM,ES). Both answers are accepted.

P(S,A,¬AF,¬EM,ES)=P(S|A)P(¬EM|A)P(A|ES,¬AF)P(ES)P(¬AF) =

0.75*0.3*0.8*0.05*0.7= 0.0063 **OR**

P(S|A,¬AF,¬EM,ES)=0.75

[1] The exam said *when no alarm sounds*, but this is not possible to compute. This was noted during the exam and changed to *when the alarm sounds*.

Task 2: Probabilistic Reasoning over Time (20p)

a) **Markov assumptions (4p):** What is a *transition model* and a *sensor model*? Explain the *Markov assumption for a second-order Markov model* and the *sensor Markov assumption* in your own words. Write the definitions of these two assumptions as formulas.

Transition model (1p): The transition model specifies the probability distribution over the latest state variables given the previous values: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1})$.
Markov assumption for seond-order Markov model (1p): The current state depends only on only on the two previous states: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2},\mathbf{X}_{t-1})$.
Sensor model (1p): The sensor model specifies the the probability distribution over the evidence variables: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t},\mathbf{E}_{0:t-1})$.
Sensor Markov Assuption (1p): The evidence variables, $\mathbf{E}_t$, depend only on the current state variables and not any of the history of state and evidence variables: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t},\mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$.

b) **Temporal Inference (8p):** Give an overview of the four basic inference tasks for temporal models.

Filtering (state estimation) (2p): Task of computing the belief state – the posterior distribution over the most recent state – given all evidence to date. This means computing: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$.
Prediction (2p): The task of computing the posterior distribution over the future state given all evidence to date. This means computing: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$.
Smoothing (2p): Task of computing the posterior distribution over a past state, given all evidence up to the present. This means computing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$, for som $0 \leq k < t$.
Most likely explanation (2p): The task of computig the squence of states that is most likely to have generated a set of observations, given a sequence of observations. This means computing $argmax_{\mathbf{x}_{1:t}}\, P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$.

c) **Filtering (8p):** We cannot measure temperatures from the past, so we have to do it indirectly. We assume that there is a correlation between tree growth rings and temperature. Figure 1 illustrates this tree growth world.

If it was a hot temperature (H) the probability of a large (L) tree growth ring was 0.8 and the probability of a large tree growth ring was 0.3 if the temperature was cold (C).

The probability that a hot year is followed by a hot year is 0.7 and the probability that a cold year was followed by a cold year is 0.6.[2]

Assume a prior of $\mathbf{P}(t_0) = <H = 0.6, C = 0.4>$.

Compute $P(H_2 \mid l_1, l_2)$. That is: Compute the probability of the weather being hot at time step 2, given that the tree ring sizes were large for time step 1 and 2.

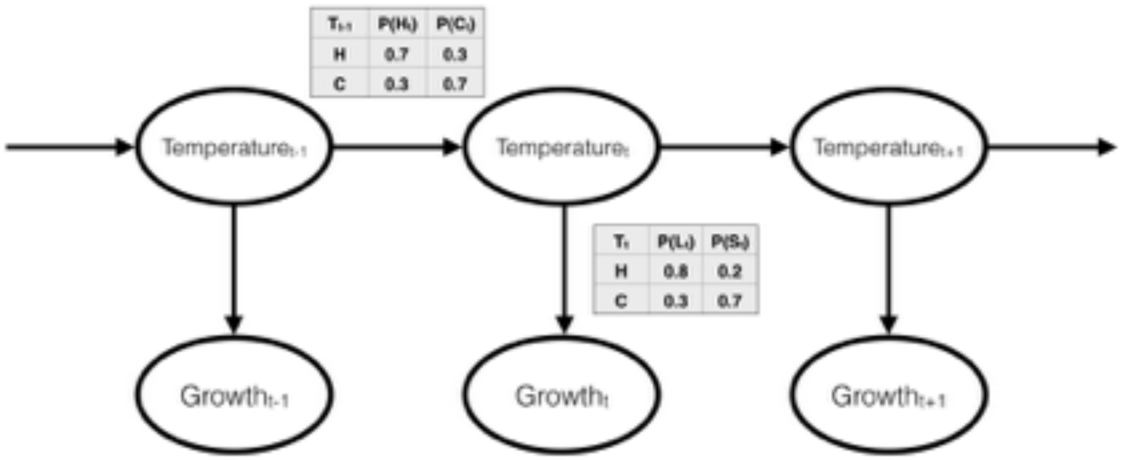


Figure 1: Bayesian network structure and conditional distributions describing the tree growth world.

Note 1: L = t => S = f and L = f => S = t **instead of** G = t or f.

Note 2: *Difference between text and figure in probabilities of cold year followed by cold year, and hot year followed by hot year.*

Numbers from Figure:

Time step 0: $P(T_0) = <0.6,0.4>$

Time step 1: Large growth ring, so L_1 is true.

$\mathbf{P}(H_1) = \sum \mathbf{P}(H_1|h_0)\mathbf{P}(h_0) = <0.7, 0.3> * 0.6 + <0.3,0.7>*0.4 = <0.54,0.46>$ (sum over h_0)

$\mathbf{P}(H_1|l_1) = \alpha\, \mathbf{P}(l_1|H_1)\mathbf{P}(H_1) = \alpha <0.8, 0.3>*<0.54,0.46> = \alpha <0.43,0.14> = <0.75, 0.25>$

Time step 2: Large growth ring, so L_1 is true.

$\mathbf{P}(H_2|l_1) = \sum \mathbf{P}(H_2|h_1)\, \mathbf{P}(h_1|l_1)$ (sum over h_1)

$= <0.7, 0.3> * 0.75 + <0.3,0.7>*0.25> = <0.6,0.4>$

$\mathbf{P}(H_2| l_1,l_2) = \alpha\, \mathbf{P}(l_2|H_2)\, \mathbf{P}(H_2|l_1) = \alpha<0.8, 0.3>*<0.6,0.4>= \alpha<0.48,0.12>=<0.8,0.2>$

Alternatively, values from text:

[2] This text differs from the figure. This was noted during the exam and the students were asked to choose one of them (and specify which they chose, but if they do not this will not affect anything).

Time step 0: $P(T_0) = <0.6,0.4>$

Time step 1: Large growth ring, so L_1 is true.

$\mathbf{P}(H_1) = \sum \mathbf{P}(H_1|h_0)\mathbf{P}(h_0) = <0.7, 0.3> * 0.6 + <0.4,0.6>*0.4 = <0.58,0.42>$ (sum over h_0)

$\mathbf{P}(H_1|l_1) = \alpha \mathbf{P}(l_1|H_1)\mathbf{P}(H_1) = \alpha <0.8, 0.3>*<0.58,0.42> = \alpha <0.46,0.13> = <0.78, 0.22>$

Time step 2: Large growth ring, so L_1 is true.

$\mathbf{P}(H_2|l_1) = \sum \mathbf{P}(H_2|h_1)\ \mathbf{P}(h_1|l_1)$ (sum over h_1)

$= <0.7, 0.3> * 0.78 + <0.4,0.6>*0.22 = <0.63,0.37>$

$\mathbf{P}(H_2|\ l_1,l_2) = \alpha\ \mathbf{P}(l_2|H_2)\ \mathbf{P}(H_2|l_1) = \alpha<0.8, 0.3>*<0.63,0.37>= \alpha<0.5,0.11>=<0.82,0.18>$

Task 3: Case-Based Reasoning (20p)

a) **CBR cycle (5p):** Illustrate the CBR cycle and explain its different components.

Input case: A description of a problem without a solution.

Retrieve: A process where past cases that are similar to the input case are retrieved from the case-base

Reuse: The step where the solutions to the past cases are adapted to the problem described by the input case. This can include copying the solution of the most similar case or complex adaptation rules for combining the solutions of several past cases.

Resolve: The process step in which the solution suggested by the CBR system is validated or invalidated by a user.

Retain: The process step in which a case is stored in the case base. This can involve just storing the case in the case-based or changing indexes in the case-base, generalize cases or more.

Case-base: A database of past problems solved in form of cases with a problem description and a solution.

b) **Case (3p):** A case is a central concept in CBR, and cases typically have two parts. Which parts are they?

Cases have a problem description and a solution.

c) **Similarity (4p):** Similarity is important in CBR. Explain what a similarity measure is and provide one or more examples.

Similarity measures are real valued functions that quantify the similarity between two objects.

Typically, the similarity is normalized between 0 and 1. The similarity can be computed between two cases (global similarity) or two features (local similarity) of the two cases. Cases could describe patients whos feature could be age and weight. Patient A has an age of 18 and weight 60kg while patient B has is 70 years old and weighs 80kg. How similar are the patients given that a similarity score of 0 is not similar at all and 1 is totally the same? Cases could be represented using the vector <age,weight> so that A=<18,60> and B=<70,80>.The global similarity between two cases could be computed as the weighted sum of the local similarity scores between all the features: $sim(case_A, case_B)=\sum w_i * sim_i(feature_{A,i}, feature_{B,i})$, where i specifies the i-th feature of the cases and w is the weight assigned to a feature. Weights specify the importance of a feature in the similarity calculation. When we have nummeric attributes, we can use vector similarity measures, such as cosine similarity:

$$cos\_sim(A,B) = A \cdot B/(||A||\ ||B||)=$$

$$(18*70+60*80)/(sqrt[18^2+60^2]*sqrt[70^2*80^2])=$$

$$(1260+4800)/(62,64*106,3)= 6060/6659=0.91.$$

The similarity between objects can be calculated using numeric functions, the distance between objects in graphs or trees, using neural networks or rules among many other possibilities.

d) **CBR System (8p):** We are going to build a CBR system for diagnosing car faults given the following information.

We have two observations:

Car A was a 2009 model of make VW Golf where the state of the lights were OK, the state of the light switch was OK, and the problem was that the front light did not work. The mechanic had to replace the front light fuse because it was defect.

Car B was a 2010 model of make VW Passat with surface damage on the lights, the light switch was OK, but the front light did not work. The mechanic had to replace the front lights and the bulb as it was broken.

Now, the mechanic gets a new observation, car C.

Car C is a 2008 model of make Volvo where the state of the lights are OK, the state of the light switch is OK, but the front light is not working.

Design a CBR system capable of diagnosing car C. Explain a possible diagnosis using the CBR cycle.

Case: Car Faults (1p)

Problem description:

- Year
- Make
- Light condition
- Light switch condition
- Problem

Solution:

- What the mechanic had to do.

Year and make are probably not very relevant so these features can be dropped or given a low weight in the similarity calculation.

Similarity measures (1p) for light condition, light switch condition and problem could be boolean functions that returns 0 or 1 based on whether the conditions are the exact same or not.

The three features found relevant are given the same importance in the global similarity calculation, that is, the combined feature similarity when comparing the cases and not only the features. The two features deemed irrelevant for case retrieval are dropped from the similarity computation.

Case-base (1p): The case-base contains the two solved cases for car A and B.

Input case (1p): The input case is car C that does not have a solution.

Retrieval (1p):

$Sim(car_A, car_C) = [$

$sim(light\_condition_A, light\_condition_C) +$

$sim(light\_switch\_condition_A, light\_switch\_condition_C) +$

$sim(problem_A, problem_C)]/3 = (1+1+1)/3 = 1$

$Sim(car_A, car_C) = (0+1+0)/3 = 0.33$

Car A is most similar.

Reuse (1p): Copy solution from case A to case C.

Solution case C: Replace the front light fuse.

Revise (1p):

Does the solution work for case C? If not change the solution to replacing the front light fuse did not work.

Retain (1p):

Store Case C with updated solution (if the solution had to be updated) in the case-base. Now the case-base contains solved cases for car A, car B and car C.

Task 4: Various

a) **Data sets (4p):** Explain the difference between training set, validation set and test set.

Training set: Data set used for training in supervised learning where each sample $<x_i,y_i>$ is comprised of of an input, x_i, and and output, y_i, and y_i is generated by an unknown function $y = f(x)$ which the machine learning method is to approximate.
Validation set: A part of the training set that can be put aside to evaluate the hypothesis while developing the hypothesis.
Test set: Data set that is distinct from the training and validation set and is used to test the accuracy of the hypothesis generated by the machine learning method.

b) **Cross validation (4p):** Explain the difference between hold out cross-validation, k-fold cross-validation and leave-one-out cross-validation.

Hold out cross-validation: Randomly splitting of the availbale data into a training set from which the learning algorithm produces the hypothesis *h* and a test set which is on which the accuracy of *h* is evaluated.
K-fold cross-validation: All examples can be used for both training the hypothesis *h* and testing it. Split data into k subsets. Then, perform *k* rounds of learning where *i/k* of the data is held out as a test set and the remaining data is used for training. The average test score of the *k* rounds should give a better estimate than a single source.
Leave-one-out cross-validation: The extreme version of *k*-fold cross-validation where *k*=*n*, and *n* is the amount of data samples.

c) **Chinese room (4p):** Explain the Chinese room thought experiment and what it is meant to illustrate. Please do not use more than two pages.

The Chinese room experiment: System: Human who only understand English, equipped with a rule book written in English, and various stacks of paper (some blank and some with indecipherable inscriptions). Human = CPU, rule book = program and stacks of paper = memory.

The system is inside a room with a small opening to the outside, through which indecipherable symbols written on paper appear. The human find matching symbols in the rule book and follows the instructions in the rule book, such as writing symbols on paper, rearranging the paper stacks and so on. Eventually, the nstrutions will cause one or more symbols to be transcribed on paper and sent out through the opening.

The system is capable of answering intelligent questions in Chinese without understainding the answers it gives.

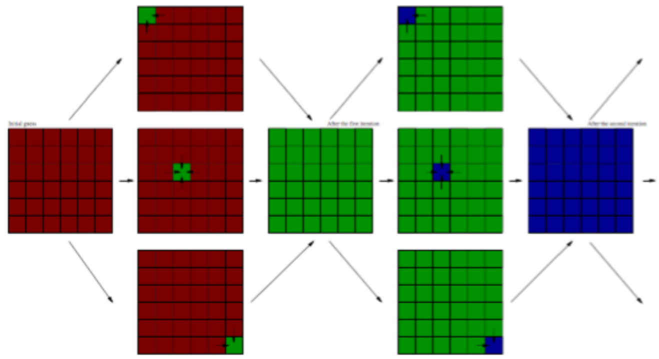
What it is meant to illustrate: A system representing a program that passes the Turing Test, but which does not understand anything of its input and outputs.

d) **Value iteration (8p):** Explain the value iteration algorithm and write pseudo code for it.

Explanation of Value iteration algorithm (4p):

Value iteration — Fix-point iterations in "value-space"

Start with an initial guess at the utility function, and iteratively refine this using the idea of fix-point iterations:



The updating function:

$$\hat{U}_{j+1}(s) \leftarrow R(s) + \gamma \cdot \max_a \sum_{s'} P(s' \mid a, s) \cdot \hat{U}_j(s').$$

Pseudo code for Value iteration algorithm (4p):

Value iteration: The algorithm

1. Choose an $\epsilon > 0$ to regulate the stopping criterion.
2. Let U_0 be an initial estimate of the utility function (for example, initialized to zero for all states).
3. Set $i := 0$.
4. **Repeat**
   1. Let $i := i + 1$.
   2. **For** each states s in S **do**

      $$\hat{U}_i(s) := R(s) + \gamma \cdot \max_a \sum_{s'} P(s' \mid a, s)\hat{U}_{i-1}(s').$$

5. **Until** $|\hat{U}_i(s) - \hat{U}_{i-1}(s)| < \epsilon(1-\gamma)/\gamma$ for all s.