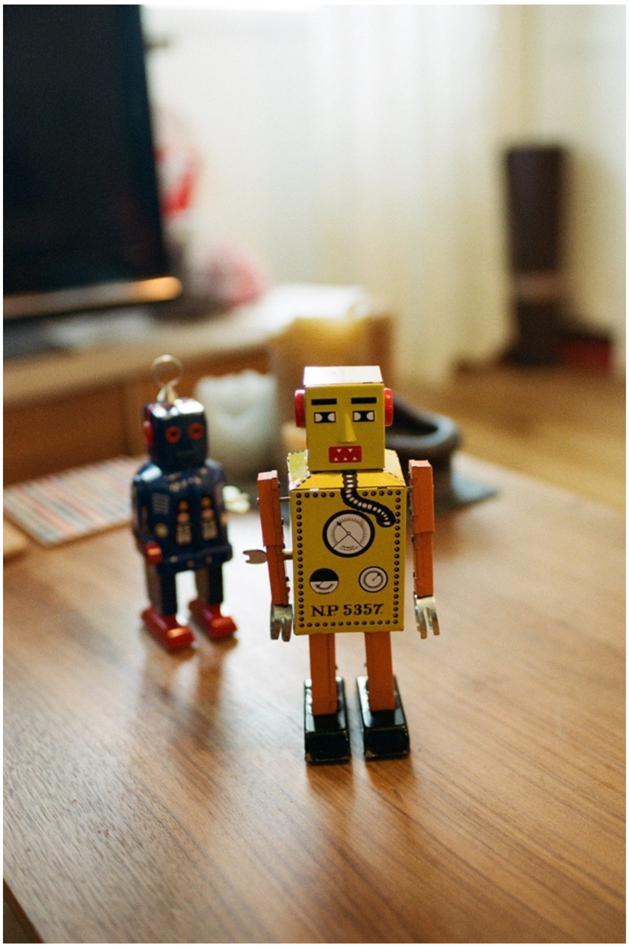


Artificial Intelligence 2, DVA265

Term: Spring 2024 (VT24)
Location: Eskilstuna
Programme: Bachelor of Science (BSc) in Applied Artificial Intelligence

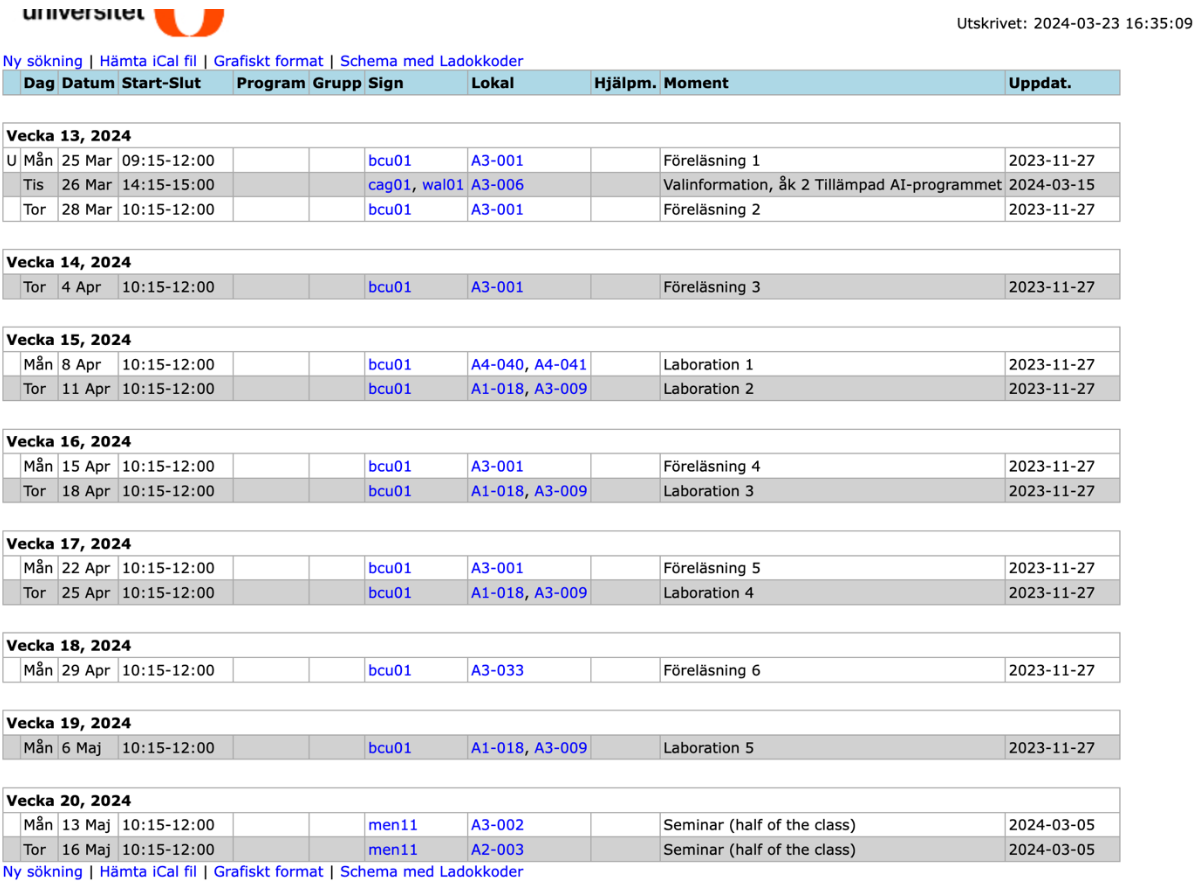
Examiner and lecturer: Baran Cürüklü, baran.curuklu@mdu.se, 073-9607453 (visiting address: U1-079 Högskoleplan 1, Rosenhill, Västerås)
Lecturer in ethics: Maria Ehn maria.ehn@mdu.se
Laboratory assistant: Elmeri Syrjänen elmeri.syrjanen@mdu.se

Document history

Version	*Date*	*Contributing person*	*Contribution*
0.1	01.03.2024	B. Cürüklü	Initial version of the document is finalised.
0.2	04.03.2024	Maria Ehn	Ethics added

1. Course schedule

This is the link to the schedule: The schedule. Please, take a look at think link to access correct information regarding our activities in the course.



Utskrivet: 2024-03-23 16:35:09

Ny sökning | Hämta iCal fil | Grafiskt format | Schema med Ladokkoder

	Dag	Datum	Start-Slut	Program	Grupp	Sign	Lokal	Hjälpm.	Moment	Uppdat.
Vecka 13, 2024										
U	Mån	25 Mar	09:15-12:00			bcu01	A3-001		Föreläsning 1	2023-11-27
	Tis	26 Mar	14:15-15:00			cag01, wal01	A3-006		Valinformation, åk 2 Tillämpad AI-programmet	2024-03-15
	Tor	28 Mar	10:15-12:00			bcu01	A3-001		Föreläsning 2	2023-11-27
Vecka 14, 2024										
	Tor	4 Apr	10:15-12:00			bcu01	A3-001		Föreläsning 3	2023-11-27
Vecka 15, 2024										
	Mån	8 Apr	10:15-12:00			bcu01	A4-040, A4-041		Laboration 1	2023-11-27
	Tor	11 Apr	10:15-12:00			bcu01	A1-018, A3-009		Laboration 2	2023-11-27
Vecka 16, 2024										
	Mån	15 Apr	10:15-12:00			bcu01	A3-001		Föreläsning 4	2023-11-27
	Tor	18 Apr	10:15-12:00			bcu01	A1-018, A3-009		Laboration 3	2023-11-27
Vecka 17, 2024										
	Mån	22 Apr	10:15-12:00			bcu01	A3-001		Föreläsning 5	2023-11-27
	Tor	25 Apr	10:15-12:00			bcu01	A1-018, A3-009		Laboration 4	2023-11-27
Vecka 18, 2024										
	Mån	29 Apr	10:15-12:00			bcu01	A3-033		Föreläsning 6	2023-11-27
Vecka 19, 2024										
	Mån	6 Maj	10:15-12:00			bcu01	A1-018, A3-009		Laboration 5	2023-11-27
Vecka 20, 2024										
	Mån	13 Maj	10:15-12:00			men11	A3-002		Seminar (half of the class)	2024-03-05
	Tor	16 Maj	10:15-12:00			men11	A2-003		Seminar (half of the class)	2024-03-05

Ny sökning | Hämta iCal fil | Grafiskt format | Schema med Ladokkoder

2. Reading instructions

We start with a few chapters (Chapter 1 that you may have read during the **Artificial Intelligence 1 (AI1)** course.

So why do I recommend you to read them again? Well, in this course our ambition is to look into **intelligent agents** and we will try to advance our knowledge in designs of intelligent agents that interact with the environment, with each other, and hence create complex interactions.

Table 1. *Reading instructions*

Chapter	Lecture	Notes
Ch. 2 Intelligent Agents	1	We will design advanced agents in this course, probably more advanced than you have done in the AI1 course. Please, read this chapter (again) with that ambition, and this about what you would like to do every assignment.

Ch. 4 Search in Complex Environments	2	**Sect. 4.1.4** Evolutionary algorithms (page 133-137). However, you will need to relate population-based algorithms to single state algorithms such as **Hill-climbing** and **Simulated annealing**.
Ch. 17 Multiagent Decision Making	3	Read and try to understand the concepts. We will blend this chapter with what we did in **LAB3** to define a problem in **LAB4**.
Ch. 11 Automated planning		Focus on **Sect. 11.1** Definition of Classical Planning and block-world problem.
Ch. 7 Logical Agents	4	Read all. Compare with what is **Chapter 2** Intelligent Agents.
Ch. 8 First-Order Logic		Again, read all, please.
Ch. 9 Inference in First-Order Logic		Read all, but focus on **Sects. 9.2** Unification, **9.3** Forward chaining, and **9.4** Backward chaining. *(In this chapter you will need the theory from the **Chapters** 7 & 8.)*
Ch. 10 Knowledge representation	5	Critical to understand all AI
Ch. 12 Quantifying Uncertainty		Read all, but focus on Bayes' Rule
Neuroscience, Computational neuroscience, Cognitive neuroscience	6	Own material. The ambition is to go through some of the theories in neuroscience, and cognitive neuroscience, from a computational perspective and see how they have contributed to AI.
Ethics – Fairness – Equality in AI-based Agents	On-line lecture material	Look through the short films. Module 1 is a basic introduction to ethics in AI. Module 2 is the focus of ethics in this course and the assignment. Read all article sections listed under this module. Module 3 presents the assignment (see section 3.2.4)
	Question time in Zoom	Teacher available for questions related to upcoming seminar (via Zoom)
	Seminar	Presentation of assignment (6 groups per occasion)

The last part of the course is about ethics (including gender equality) related to AI-based conversational agent. It contains:

(I) On-line course material

Module1 Basic concepts (ethics-fairness-equality) and related literature

Recommended reading

- European Commission: High-Level Expert Group on AI presented Ethics Guidelines for Trustworthy Artificial Intelligence. Ethics Guidelines for Trustworthy AI. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai [accessed Feb 19th 2024]. *Ethical principles described on pp. 11-13*

- Jobin, A., Ienca, M., Vayena, E.: The global landscape of ai ethics guidelines. Nature Machine Intelligence pp. 1–11 (2019). *Ethical principles described on pp. 7-13*

Module2Ethical concerns particularly relevant for agent-based systems (specific focus: conversational agents)

Reading instructions

- At least Sections 3 and 4 *in*

Ruane, E., Birhane, A., & Ventresque, A. (2019, December). Conversational AI: Social and Ethical Considerations. In AICS(pp. 104-115). https://ceur-ws.org/Vol-2563/aics_12.pdf

- pp 285-287 *in*

Luxton DD. Ethical implications of conversational agents in global public health. Bull World Health Organ. 2020 Apr 1;98(4):285-287. doi: 10.2471/BLT.19.237636. Epub 2020 Jan 27. PMID: 32284654; PMCID: PMC7133471. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7133471/

- pp 106-116 *in*

West, M., Kraut, R., Chew, H.E.: I'd blush if I could: closing gender divides in digital skills through education (2019). https://unesdoc.unesco.org/ark:/48223/pf0000367416

Module3 The assignment

Reading instructions: Same as in module 2. See (see section 3.2.4)

(II) Teacher available for questions related to upcoming seminar (via Zoom)

(III) Seminar with presentations of assignment

3. Assignments

You will solve the assignments in groups of two (2) students, or if you prefer alone, meaning by yourself. Remember that LAB1-4 is your examination, so please minimize you interaction with your class mates regarding the specific details of your solutions, simply stick to your own group!

LAB2 and LAB4 provides you options regarding the problem to be solved (see below, please).

The assignments aim at covering symbolic and non-symbolic AI. Symbolic Ai is based on, or derived from **logic**, whereas non-symbolic AI is about **numerical calculations** (some would say number crunching).

Note that symbolic versus non-symbolic is about how to represent the world, or the problem. So, the question of representing a problem to fit an algorithm is important. **Knowledge representation** is a key activity in AI, and actually in all problem solving. Simply put, you don't use a hammer for everything when you renovate your kitchen!

In the course schedule we have 5 supervision meetings for supporting you with the assignments. These are **8th, 11th, 18th, and 25th of April**, followed by **6th of May**.

As you can see below we will start with numerical AI, that is LAB 3-4, after that we will look at symbolic AI in LAB1-2.

Table 2. *Overview of the assignments, which also corresponds to the course examination.*

	Type of representation	**Credits (hp)**	**Scheduled**	**Note**	**Deadline for the 'LAB'**
LAB1	Symbolic	0,5	*April 8, 11, 18, 25, and May 6*	We start with **LAB3-4** (numerical AI) and continue with **LAB1-2** (symbolic AI).	*Tuesday 23rd May
LAB2		2,5			
LAB3	Numerical	2,0			*Tuesday 2nd of May
LAB4		2,5			

*You will be able to do present your results online (Zoom/Teams), in addition to the schedules sessions. Try to see these deadlines as hard so that they can guide.

3.1. EA and optimisation (LAB 3)

Implement the most simple and classic genetic algorithm. Implement selection, crossover, mutation of your choice. The genes will take binary values, so they can be either '0' or '1'. In a sense these individuals represent yes/no answers and this is very much symbolic representation as well.

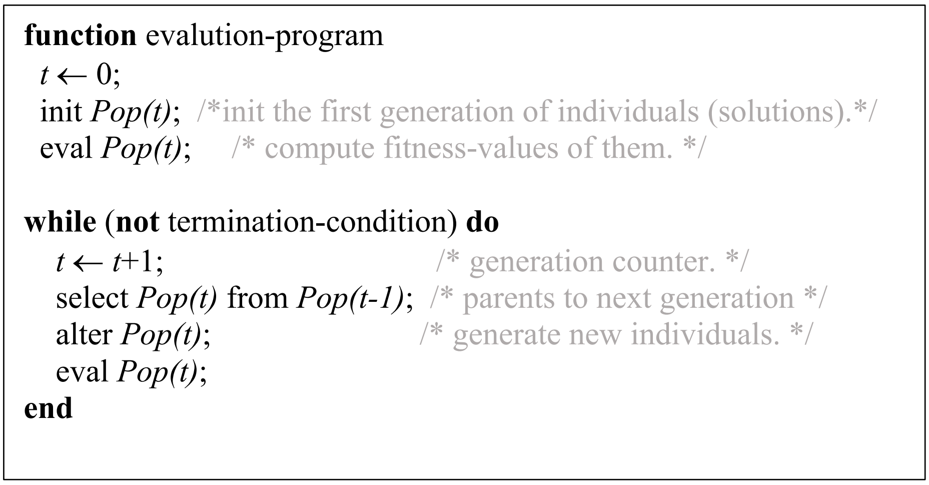
In this test number of genes of the individuals' chromosome should be 50:

Individual _1: [0 0 1 … 1]. <- *you see only 4 out of 50 genes in the chromosome*
Individual _2: [0 1 1 … 0]
….
Individual _20: [1 1 1 … 0]

Define now **an objective function** to "Maximise the sum of all genes", well this means that the fitness function will try to find an individual with the chromosome consist of only '1'

This is how the **best individual** should look like [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1].

Q1: Do you need crossover and mutation?
Q2: What probabilities for crossover and mutation seems to be working well?
Q3: Test your algorithm with different population sizes, from 10 to 100 individuals. Which population works well, and why? How does the performance of the algorithm changes with the population size, with respect to (1) time to convergence, (2) memory requirements?



```
function evalution-program
  t ← 0;
  init Pop(t);  /*init the first generation of individuals (solutions).*/
  eval Pop(t);     /* compute fitness-values of them. */

while (not termination-condition) do
    t ← t+1;                          /* generation counter. */
    select Pop(t) from Pop(t-1);  /* parents to next generation */
    alter Pop(t);                    /* generate new individuals. */
    eval Pop(t);
end
```

Figure 1. *The pseudo code of a genetic Evolutionary algorithm.*

3.1.1. Hints for the implementation

Remember this pseudo code from the 2nd lecture (Fig.1), and the other slides on this topic. What you will do is the following:

1. Implement a roulette wheel **selection algorithm**.
2. The **fitness (or objective) function** is very simple. The goal is to get all genes = 1 for the best solution. Thus, start with just adding all genes in an individual's chromosome to get the total sum.
    a. **Afterwards, try to improve the fitness function. Can you do that?**
3. The whole current population will be replaced after selection → crossover → mutation. Thus, if you have 50 individuals in the population, that population number will never change.
4. It is OK if one individual is selected several times. Actually, the roulette wheel algorithm will result in this behaviour. It is also OK if a pair of individuals are the same individual (this is not a good thing, however, remember that testing this will mean more computations so we ignore it).
5. As in the slides implement a **crossover algorithm** that with a probability of 60% swaps the genes.
    a. **Afterwards, you will need to test other values**, e.g., 10%, 20%, 40%, and 80% just to see if you can speed up the search.
6. Mutation is again very simple. With a probability of 3% flip '0' → '1', or '1' → '0'.
    a. **Afterwards, change the mutation value** to, very low = 1%, or much higher = 5%, or 10%. What happens with the performance (in convergence time)?

7. Creating the new population:
    a. When a pair of parents are modified (with crossover and mutation) 2 new offspring are created. Thus, you have 4 individuals in total. Rank them based on their fitness values and add the best 2 into the new population.

With the alternatives test on the fitness function (Nr. 2), crossover (Nr. 5), and mutation (Nr. 6) you have tested different algorithms, for this simple problem. **What are your conclusions?**

3.2. EA and artificial agents for planning (LAB4)

3.2.1. Background

The ambition is to start from the algorithm in LAB3 and scale it up and implement a platform for interactions between agents and at the same time solve a planning problem (see Sect. 3.2.1). Remember the EA-operators that you have implemented in LAB3: (i) selection, (ii) crossover and (iii) mutation.

(i) **Selection:** This operation is about competition between the agents. They compete to have the possibility to produce their own offspring (children).

(ii) **Crossover:** This operator is about exchange of information between the agents. Thus, this is where agents interact with each other.

(iii) **Mutation:** In this step you can think about random changes in an agent status (or know

In Sect. 3.2.1. you have the framework for the problem you will solve. You need to discuss the details with your group member (and with your class mates, at this stage you all can talk to each other freely) how you will implement the problem. Get back to me with your ideas before starting the implementation.

This assignment will help you to think about all the intricates of designing a simple, yet powerful, MAS world.

Learning outcome 8 is on "carrying out an ethical and gender equality analysis of an agent-based AI system". In Sect. 3.2.4 the assignment on ethics and gender equality in agents is presented.

3.2.2. The Multi-agent system (MAS) world

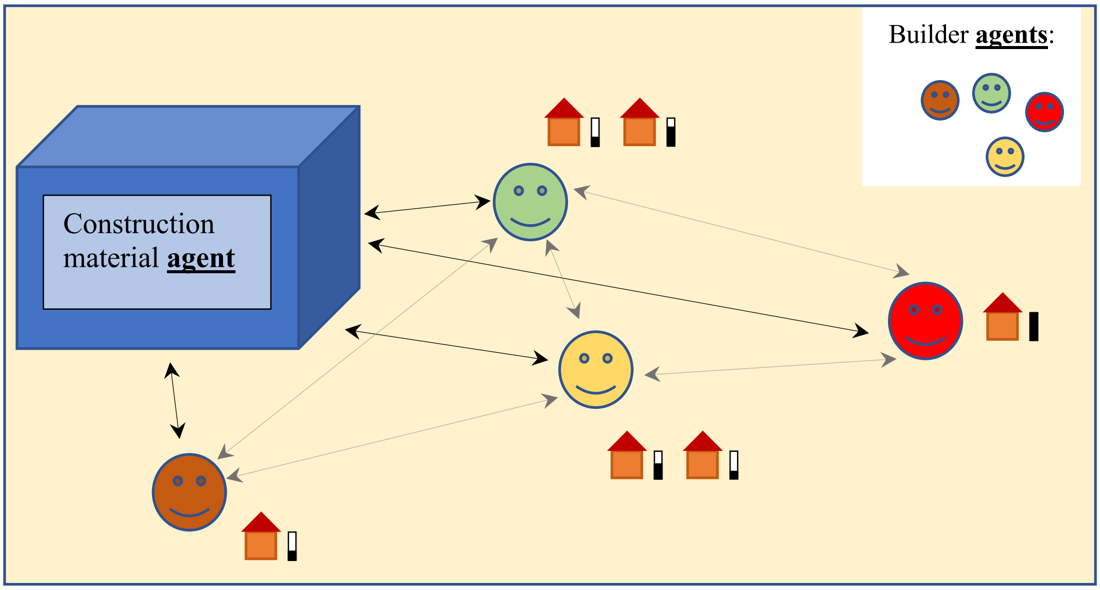


Figure 2. *An illustration of the MAS world that you will implement. There is one (1) Construction material agent. This agent sells all the material that is needed for building a house. The builder agents build houses, which they sell, and by ding that earn money. The agent that has built the most (or earned the most) will win the competition.*

In the MAS world there are 2 different types of agents:

- 1 Construction material agent
- Builder agents.

The **builder agents** compete with each other to build the maximum number of houses. They can start building maximum 2 houses at the same time. All the building material will be purchased from the **construction material agent**.

All houses consist of one floor, and one garret (swe: vindsrum). The MAS world house is, of course, an oversimplification (Table 3. has all the details).

Table 3. Requirements (the components) for building a house in the MAS world.

Output	The components
The house	1 floor + 1 garret
1 floor	4 bed rooms, 2 bath rooms, one living room
1 bed room	2 windows + 1 door + 1 wall module
1 bath room	1 door + 1 toilet-seat + 1 tab + shower cabin + 1 wall module
1 living room	1 door + 3 windows + 1 wall module
1 hall	1 outside-door + 1 window + 1 wall module
1 garret	3 windows + 1 door + 1 wall module

As you can see you need to fulfil certain requirement for building a house, thus building a house can be formulated as a planning problem. Also, a floor (and a garret) requires components also. This is not a hard planning problem, however, it is still a planning problem.

Table 4. The cost of the components

The component	The price (SEK)*	
1 door	2500,-	Inside door for the rooms and the toilet, etc.
1 outside-door	8500,-	
1 window	3450,-	All windows are the same.
1 wall-module	75000,-	A 4-wall module making a room
1 toilet-seat	2995,-	
1 tab	2350,-	
1 shower cabin	8300,-	

* The prices are realistic and come from https://www.bauhaus.se/

3.2.3. Completing the problem definition and finalising the implementation

There are a few details that are remaining in the problem definition. Discuss them with your group member and finalise the design:

1. The construction material agent cannot have unlimited material in any given time. How should a realistic agent behave?
2. When 2 builder agents compete to purchase, there must be a way for the construction material agent to choose between them. what should that criterion be?
3. You need to think about how often a builder agent tries to swap material with other agents.
4. There are probably other details as well…

3.2.4. Ethics and gender equality in agents

This assignment is related to the course's learning outcome 8 (i.e.," after completing the course, the student shall be able to carry out an ethical and gender equality analysis of an agent-based AI system"). The task is focused on *AI-based conversational agent*.

Content and realization:
The assignment includes:

(I) Reading in three articles on ethical concerns related to AI-based conversational agent (individual work):

- at least Sections 3 and 4 *in*

Ruane, E., Birhane, A., & Ventresque, A. (2019, December). Conversational AI: Social and Ethical Considerations. In AICS(pp. 104-115). https://ceur-ws.org/Vol-2563/aics_12.pdf

- pp 285-287 *in*

Luxton DD. Ethical implications of conversational agents in global public health. Bull World Health Organ. 2020 Apr 1;98(4):285-287. doi: 10.2471/BLT.19.237636. Epub 2020 Jan 27. PMID: 32284654; PMCID: PMC7133471. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7133471/

- pp 106-116 *in*

West, M., Kraut, R., Chew, H.E.: I'd blush if I could: closing gender divides in digital skills through education (2019). https://unesdoc.unesco.org/ark:/48223/pf0000367416

- You are welcome to also add references from own searches.

(II) Analyzing a conversational agent related to ethics and gender equality based on aspects from articles (work in groups of three student):

1. Select an existing AI-based conversational agent (text- or voice based)
2. Analyze the system in relation to ethics and gender equality by

a) Selecting 4-6 aspects related to ethics (including gender equality) from the articles that you find relevant to analyze for your system

b) Exploring/interacting with the system and discussing in the group

The analysis should consider risks for individual users (including men and women, also from vulnerable groups) and risks on a societal level. Each identified risk should be briefly described based on the articles and based on your interactions with the system.

(III) Preparing a 10 min presentation where you demonstrate

1. The system
2. The aspects related to ethics (including gender equality) that your analysis is based on
3. The results of your analysis illustrated by your interactions with the system.

Your interactions with the system can be presented "live" or via screen shots on PPT: s.
Please note that all group members need to actively contribute to the analysis and the presentation (both the preparations and the demonstration at a seminar described below).

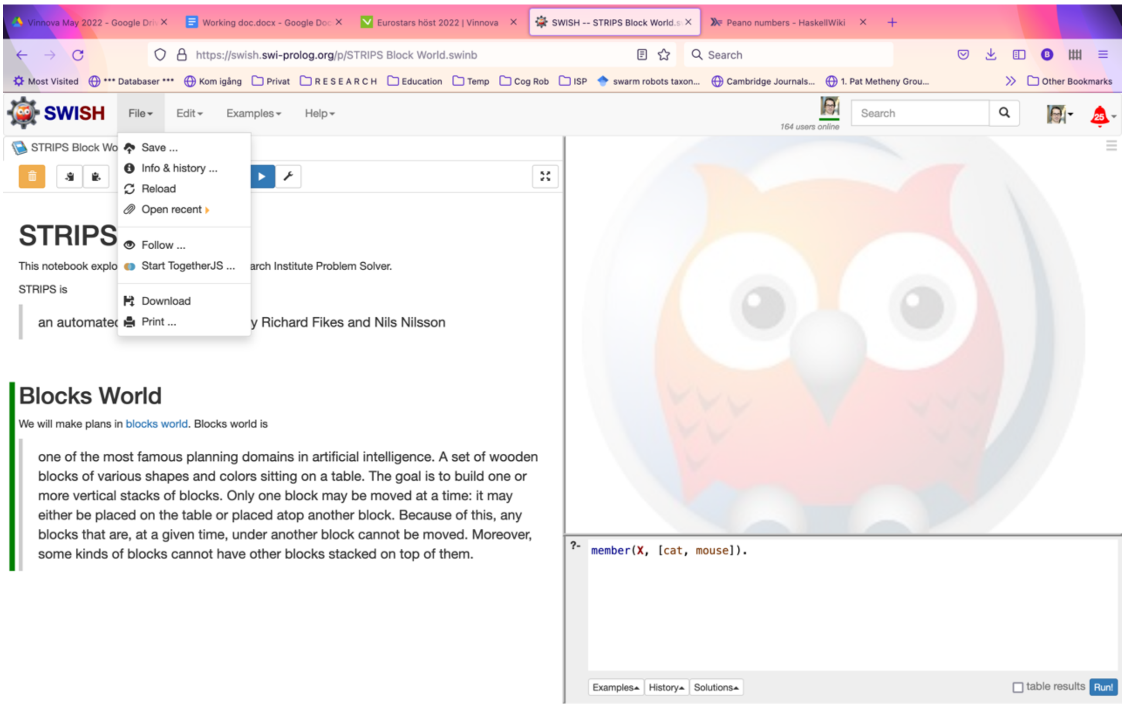
The teacher is available for questions related to upcoming seminar (via Zoom) on May 7th at 14.00-15.00.

Examination:
Will take place in a seminar where the group present their work for the class. Active participation in the group presentation and discussion is required for grade "pass" (G) on the assignment. Student that are absenr from or insufficiently active in the seminar will receive an individual written assignment from the teacher (after the seminar).
Seminar occasions (both including presentations from 6 groups) are:
Monday May 13th (at 10-12) or Thursday May 16th (at 10-12). Both occasions will be in Eskilstuna.
Please note that all groups need to sign up for one of seminar slot in Canvas ("first come, first serve") before May 7th.

3.3. Introduction to logic programming (LAB1)

This assignment is about using logic to represent information. This is called symbolic AI. **Why is it symbolic?** Well, logic is about symbols and manipulation with symbols. Symbolic AI is the most classical form om AI. The Prolog language have been around for a long time, and it is an excellent tool in representing information in propositional (swe: satslogik) and predicate (swe: predikat) logic.

You will use the web implementation of the SWI Prolog: https://swish.swi-prolog.org/p/STRIPS%20Block%20World.swinb You can open new Program editor using the '+' sign. **This is community, so there are excellent examples/tutorial**, and you can run Prolog from your browser!



https://swish.swi-prolog.org/p/STRIPS Block World.swinb

STRIPS

This notebook explo … arch Institute Problem Solver.

STRIPS is

> an automatec … y Richard Fikes and Nils Nilsson

Blocks World

We will make plans in blocks world. Blocks world is

> one of the most famous planning domains in artificial intelligence. A set of wooden blocks of various shapes and colors sitting on a table. The goal is to build one or more vertical stacks of blocks. Only one block may be moved at a time: it may either be placed on the table or placed atop another block. Because of this, any blocks that are, at a given time, under another block cannot be moved. Moreover, some kinds of blocks cannot have other blocks stacked on top of them.

```
?- member(X, [cat, mouse]).
```

Figure 3. *The SWISH page. At top right you can see a picture of me. By one click I connected my Google account/email to SWISH so I have an account now for saving files, interacting with others, etc.*

There several other tutorials, etc. available. Some good examples are here:

1. A very good tutorial on lists: https://www.cpp.edu/~jrfisher/www/prolog_tutorial/2_7.html
2. Tutorial on simple programs: https://www.tutorialspoint.com/prolog/prolog_basic_programs.htm

Write this simple program (see the blue textbox below) that consist of one predicate called `medlem/2`. This predicate is actually implemented already in all Prolog systems as the predicate `member/2`.

```
medlem(X,[X|R]).
medlem(X,[Y|R]) :- medlem(X,R).
```

This predicate defines the relationship between two entities, where the second is a list. What these 2 lines say is as follows:

- *'X' is the first element of the list, which means that it is a member.*
- *'X' is not the first member of the list, thus it is perhaps somewhere in the rest of the list. Thus, the interpreter removes the first element and tries to find 'X' is in the rest of the list called 'R'.*

Q1: Ask now the following questions to the Prolog interpreter:

```
medlem(2,[1,2,3]).
```

What is the answer?

Even this simple predicate can be used in many different ways. This shows the strength of logic programming and Prolog. See in **Q2** and beyond how we use this predicate in different ways.

Q2: Imagine that the list consists of band names. Now ask the following 2 questions. Can you say in plain language what these questions mean?

```
medlem(abba,[roxette, kiss, abba]).
```

What is the answer?

Note that we use the same predicate for both data types.

```
medlem(justin_bieber,[roxette, kiss, abba]).
```

What is the answer?

Q3: You can even mix different types:

```
medlem(”van halen”,[1, roxette, 4, ”motley crue”, 3, ”van halen”]).
```

What is the answer?

Q4: We can make the list a bit more complex. The members of the list are band names and the members of the band. Note that we cannot have predicates inside other predicates (at least not as we have done below). Let us not ask this question where abba() is the first element.

```
member(abba(M1, M2, M3,M4), [roxette(marie, per), van_halen(david,
eddie, alex, anthony),abba(agneta, anni-frid, bjorn, benny)]).
```

What is this question about? What is the answer?

Q5: Benny is a member of the band ABBA. Now ask this question where only Benny is mentioned.

```
abba(benny, X, Y, Z).
```

Did it answered by matching X, Y, Z with the other 3 members or did it say ′No′? Why?

Q5 demonstrates **unification**, which is Prolog's way of doing **pattern matching**. This means that you need to know the order of the arguments, and that Prolog does unification by comparing the arguments one by one.

Q6: Write a program that returns the difference between the largest and the smallest element in the list. Start from the predicate ′largest_element′ which is written below. Use only preicatas that you have written.

```
largest_element(X, [X]).
largest_element(X, [X|Rest]):- largest_element(Y, Rest), X >=Y.
largest_element(N, [X|Rest]):- largest_element(N, Rest), N > X.
```

Q7: Now you will test a few more Prolog predicates on your own. These are already implemented. Try to combine them to create one program. Perhaps you can find something interesting from the AI1 course. Some of the important predicates in Prolog are: `concat/3, length/2, reverse/2, sum/2, mean/2`, etc.

During the LAB3 presentation you will need to explain the program that you have decided to implement.

Q8: Peano numbers are simply a way to represent natural numbers, [0, 1, 2, 3, …] as functions starting form zero. Define now +, -, /, x for 2 Peano numbers.

Natural number	
0	0
1	f(0)
2	f(f(0))
3	f(f(f(0)))
…	…

During the LAB3 presentation you will need to explain how +, -, /, x are implemented.

3.4. Relationships and expert systems in Prolog (LAB2)

Let us now design an expert system. Usually, expert systems are based on forward chaining, however, you can choose to implement an expert system similar to how Prolog works, i.e., using the Prolog's backward chaining mechanism. This implementation decision (backward, instead of forward chaining) is yours.

My problem suggestion: In backward chaining implement the classical problem of "The Bird Identification System" described in Chapter 2 of the book Building Expert Systems in Prolog, by Dennis Merritt (the book as a PDF is in the course's Canvas page under Kursinformation).

Read mainly Sect. 2.1, and parts of Sect. 2.2 for simple interaction. Study also the Figure 2.1. It helps to visualise the problem domain. You don't need to implement any user interface, or any menu. Just make sure that you can interact with the program at a basic level.

Q1: Add now one more bird category at the level of 'laysun_albatross', 'black_footed_albatross', and 'rumpeter_swan'. You can choose a Swedish/European bird, or any other bird from the American continent.

Remember that you task is to implement an expert system, think about the current expert system and decide if you can add a Swedish, or European, bird easily, or if the current implementation only allows adding American birds.

Q2: Test the predicate `trace` to see how Prolog behaves in Q1 in LAB1 and also the examples in LAB 2. Spend some time on this task, please. Try to learn how Prolog behaves.

Q3: Test the meta predicates `fidnall`, `bagoff`, and `setof`, described in the SWI-Prolog https://www.swi-prolog.org/pldoc/man?section=allsolutions As you know, it is not possible to have predicates as inputs (terms) to other predicates, however, these **meta** predicates are exception to that rule.

Q4: For the problem in Q1, use the predicate trace to demonstrate how unification works in Prolog. **Write a report on ½ A4 of text.** Include screen dumps, as figures, to help you with the text. Imagine that receiver of this report is a 1st year computer science student.

4. Overview of the examination

***** Examination (svenska) *****

LAB1, en programmeringsuppgift som ska demonstreras, 0,5 hp, examinerar lärandemål 1 och 7, betyg Underkänd (U) eller Godkänd (G).

LAB2, en programmeringsuppgift som ska demonstreras, och presenteras med en rapport, 2,5 hp, examinerar lärandemål 1, 3, 4, 6 och 7, betyg Underkänd (U) eller Godkänd (G).

LAB3, en programmeringsuppgift som ska demonstreras, och presenteras med en rapport, 2 hp, examinerar lärandemål 2, 6 och 7, betyg Underkänd (U) eller Godkänd (G).

LAB4, en programmeringsuppgift som ska demonstreras, och presenteras med en rapport, 2,5 hp, examinerar lärandemål 2, 5, 6, 7 och 8, betyg Underkänd (U) eller Godkänd (G).

För slutbetyg Godkänd (G) krävs betyget Godkänd (G) i alla fyra laborationerna.

***** Examination (English) *****

LAB1, an assignment that is demonstrated to the teacher, 0.5 credits, examines the learning outcomes 1 and 7, marks Fail (U) or Pass (G).

LAB2, an assignment that is presented with a report and a demonstration to the teacher, 2.5 credits, examines the learning outcomes 1, 3, 4, 6 and 7, marks Fail (U) or Pass (G).

LAB3, an assignment that is presented with a report and a demonstration to the teacher, 2 credits, examines the learning outcomes 2, 6 and 7, marks Fail (U) or Pass (G).

LAB4, an assignment that is presented with a report and a demonstration to the teacher, 2.5 credits, examines the learning outcomes 2, 5, 6, 7 and 8, marks Fail (U) or Pass (G).

For final grade Pass (G) the mark Pass (G) is required in all four laborations.

4.1. Learning outcomes

Lärandemål (på svenska)

1. analysera och definiera sats samt predikatlogik för implementation av agentmodeller,
2. förklara och tillämpa populationsbaserade agentmodeller med utgångpunkt från evolutionära
algoritmer,
3. analysera och definiera de mer grundläggande metoderna inom oskarp logik, osäkerhet, och resonemang,
4. förklara och tillämpa expertsystem för att lösa ett domänspecifikt problem,
5. förklara och tillämpa planering genom populationsbaserade agentmodeller,

6. förklara hur olika representationer av ett problem med avseende på ökad prestanda kan bedömas, där representationen sker i form av antigen (i) logik, m.a.o. symboliskt, (ii) numerik eller (iii) i kombination av dessa,
7. analysera och definiera ett givet problem, samt bestämma om det kan lösas med en av de tekniker som ingår i denna kurs samt
8. genomföra en etik- och jämställdhetsanalys av ett agentbaserat AI_system.

Learning outcomes (in English)

After completing the course, the student shall be able to:

1. analyse and define propositional and predicate logic; and demonstrate how these theories can be used in logic programming for solving problems and representation of agent models,
2. explain and apply population-based agent models, such as evolutionary algorithms and its variations,
3. analyse and define the most representative methods in uncertainty, fuzzy logic, and reasoning,
4. explain and apply expert systems for solving domain specific problems,
5. explain and apply planning through population-based agent models,
6. explain the means for knowledge representation with respect to performance, especially in the context of different representation paradigms, such as (i) logic, that is symbolic, (ii) numeric, and (iii) the combination of both,
7. analyse and define a given problem with the ambition of deciding if it can be addressed by the methods covered in this course and also
8. carry out an ethical and gender equality analysis of an agent-based AI system.