F29AI – Artificial Intelligence and Intelligent Agents

Coursework 1 – Part 1 - Knowledge Representation

You should complete this coursework **individually**. This part of Coursework 1 is worth **5%** of your overall F29AI mark. Details of what you should do and hand in, and how you will be assessed, are described below.

Problem description

A popular fictional monster battling game features a type system that is used to determine the effectiveness of a monster's ability to attack or defend against another monster. In this coursework you will write a **Prolog program** to represent knowledge about this system and to answer queries using the resulting knowledge base.

The version of the game we will use includes five **monsters**: `charmander`, `bulbasaur`, `squirtle`, `pikachu`, and `eevee`. Each monster can be one of five basic **types**: `fire`, `grass`, `water`, `electric`, and `normal`. Each monster has four abilities that it can use. Each **ability** is also assigned one of the basic types. The details of each monster, its type, its abilities, and the ability types are given in the following table:

Monster	**Monster type**	**Ability**	**Ability type**

`charmander`	`fire`	`scratch`
		`fireFang`
		`firePunch`
		`thunderPunch`
			`normal`
			`fire`
			`fire`
			`electric`

`bulbasaur`	`grass`	`tackle`
		`vineWhip`
		`razorLeaf`
		`solarBeam`
			`normal`
			`grass`
			`grass`
			`grass`

`squirtle`	`water`	`tackle`
		`waterPulse`
		`aquaTail`
		`bodySlam`
			`normal`
			`water`
			`water`
			`normal`

`pikachu`	`electric`	`thunderPunch`
		`surf`
		`grassKnot`
		`thunderbolt`
			`electric`
			`water`
			`grass`
			`electric`

`eevee`	`normal`	`rainDance`
		`sunnyDay`
		`bite`
		`lastResort`
			`water`
			`fire`
			`normal`
			`normal`

E.g., `charmander` is a `fire` type monster with four abilities: `scratch` (a `normal` type ability), `fireFang` (a `fire` type ability), `firePunch` (a `fire` type ability), and `thunderPunch` (an `electric` type ability).

The **effectiveness** of a monster's ability when used on another monster depends on the ability type (of the monster using the ability) and the monster type (of the monster the ability is being used on). Certain abilities are `super` effective against certain types of monsters while other abilities are `weak` against other monster types. The effectiveness of an ability type against a monster type is represented in the following table:

`ability \ monster`	`fire`	`water`	`electric`	`grass`	`normal`
`normal`	`ordinary`	`ordinary`	`ordinary`	`ordinary`	`ordinary`
`fire`	`weak`	`weak`	`ordinary`	`super`	`ordinary`
`water`	`super`	`weak`	`ordinary`	`weak`	`ordinary`
`electric`	`ordinary`	`super`	`weak`	`weak`	`ordinary`
`grass`	`weak`	`super`	`ordinary`	`weak`	`ordinary`

E.g., a `fire` type ability is `weak` against `water` type monsters but a `water` type ability is `super` against `fire` type monsters. Combinations that aren't `super` or `weak` are `ordinary`.

What to do

Write a Prolog program to represent the knowledge in the monster game according to the following specification:

1. Encode information about the monsters and their abilities using Prolog **facts** of the following form:
   - `type(t)`: to represent the idea that `t` is a basic type.
   - `monster(m,tm)`: to represent the idea that `m` is a monster of type `tm.`
   - `ability(a,ta)`: to represent the idea that a is an ability of type `ta.`
   - `monsterAbility(m,a)`: to represent the idea that monster `m` has an ability `a.`
2. Encode effectiveness information using Prolog **facts** of the form type`Effectiveness(ta,tm,e):` an ability of type `ta` has effectiveness e against monsters of type `tm`.
3. Define a Prolog **rule** called `abilityEffectiveness(A,M,E)` to represent the idea that E is the effectiveness of an ability A against a monster M. A, M, and E should be variables in your rule definition.
4. Define a Prolog **rule** called super`Ability(M1,A,M2)` to represent the idea that ability A is a `super` effective ability for monster M1 to use against monster M2. M1, A, and M2 should be variables in your rule definition.
5. Define a Prolog **rule** called `typeAbility(M,A)` to represent the idea that ability A is an ability of monster M and that M and A have the same type. M and A should be variables in your rule definition.
6. Define a Prolog rule called `moreEffectiveAbility(A1,A2,T)` to represent the idea that ability A1 is more effective than ability A2 against monsters of type T: `ordinary` is more effective than `weak`, and `super` is more effective than `ordinary` and `weak`. A1, A2, and T should be variables in your rule definition.
7. Define a Prolog **rule** called `counterAbility`(M1,A1,M2,A2) to represent the idea that if monster M1 performs ability A1 and monster M2 performs ability A2 that A2 is a more effective than A1: `ordinary` is more effective than `weak`, and `super` is more effective than `ordinary` and `weak`. M1, A1, M2, and A2 should be variables in your rule definition.

NOTE: For parts 3-7 of the specification, ensure that you write Prolog rules. You should **not** implement Prolog facts as your solution for these parts and you will lose marks if you do this. However, it is okay if need to write multiple rules for each definition. If you are interested, the game mechanics in this coursework are based on information from https://pokemondb.net/.

What to hand in

- **Prolog program file:** Submit a single Prolog program file with all your Prolog facts and rules. Ensure that the file is a plain text file with the name `monster.pl`. Make sure there are comments in your program file to describe the different parts of your program

- **Output file:** Test your program by selecting a series of Prolog **queries** to demonstrate the various facts and rules you have implemented. Capture the queries and output to a text file. Include at least 10 queries in your output file. Ensure that the file is a plain text file with the name `output.txt`.

Deadlines

The deadline for submitting Coursework 1 (all parts) is **Thursday, 29 October 2020**. Submissions are due by **3:30pm (Edinburgh local time)** for the Edinburgh Campus, **5:00pm (Dubai local time)** for undergraduate students at the Dubai Campus, and **11:59pm (Dubai local time)** for postgraduate students at the Dubai Campus. Submit your coursework on Vision in the **Coursework** section of F29AI.

Assessment

This coursework will count towards **5%** of your overall course mark for F29AI and will be marked out of **5 marks**. You will be assessed on the correctness of your Prolog program and the output it produces.

0 **None**	**1** **Poor**	**2** **Fair**	**3** **Good**	**4** **Very good**	**5** **Excellent**
No source code or minimal source code submitted.	Weak solution. Many important requirements of the specification or important parts of the output missing. Comments missing.	Adequate solution. Some requirements of the specification implemented but important parts missing. Output cases do not provide complete coverage and/or missing many comments.	Thorough solution. Majority of program meets specification. Some small problems with the program, missing cases in the output file, and/or missing comments.	Very thorough solution. Program almost works perfectly. There might be very small problems with the program, some missing cases in the output file, and/or missing comments.	Program works perfectly. Output illustrates all important cases. Comments in code are descriptive.

Additional notes

This is an **individual coursework** assignment. Both the program and output will be checked for plagiarism. You are responsible for ensuring that your submitted program conforms to the naming conventions of facts and rules. The marker should not need to make any changes to the submitted program file to get it to work. Please ensure that your output file is readable as a plain text file.

Feedback

Individual written feedback will be provided to students approximately three working weeks after the submission of Coursework 1.

Learning Objectives

This coursework is meant to contribute to the following high-level aims for F29AI:

- To introduce the fundamental concepts and techniques of AI, including knowledge representation.
- To develop skills in AI programming in an appropriate language.

It is also meant to contribute to the following specific learning objectives for the course:

- Critical understanding of traditional AI problem solving and knowledge representation methods.
- Use of knowledge representation techniques (such as predicate logic).
- Broad knowledge and understanding of the subfields and applications of AI.
- Detailed knowledge of a subfield of AI and ability to apply its formalisms and representations to small problems.
- Practice in the implementation of simple AI systems using a suitable language.
- Identification, representation and solution of problems.
- Research skills.

Late submission of coursework

Coursework deadlines are fixed and individual coursework extensions will not be granted. Penalties for the late submission of coursework follow the university's policy on late submissions:

- The mark for coursework submitted late, but within 5 working days of the coursework deadline, will be reduced by 30%.
- Coursework submitted more than 5 working days after the deadline will not be marked.
- In a case where a student submits coursework up to 5 working days late, and the student has valid mitigating circumstances, the Mitigating Circumstances policy will apply. Students should submit a Mitigating Circumstances application for consideration by the Mitigating Circumstances Committee.

The MACS School policy on coursework submission is that the **deadline for coursework submissions**, whether hard-copy or online, is **3:30pm (Edinburgh local time)** for the Edinburgh Campus and **5:00pm (Dubai local time)** for the Dubai Campus. The University Policy on the Submission of Coursework can be found here:
https://www.hw.ac.uk/services/docs/learning-teaching/policies/submissionofcoursework-policy.pdf

Mitigating Circumstances (MC)

There are circumstances which, through no fault of your own, may have affected your performance in an assessment (exams or other assessment), meaning that the assessment has not accurately measured your ability. These circumstances are described as **mitigating circumstances**. You can submit an application to have mitigating circumstances taken into account. Full details on the university's policies on mitigating circumstances and how to submit an application can be found here:
https://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm

Plagiarism

"**Plagiarism** is the act of taking the ideas, writings or inventions of another person and using these as if they were your own, whether intentionally or not. Plagiarism occurs where there is no acknowledgement that the writings, or ideas, belong to or have come from another source." (Heriot-Watt University Plagiarism Policy). This coursework must be completed independently:

- Coursework reports must be written in a student's own words and any submitted code (e.g., PDDL) in the coursework must be your own code. Short sections of text or code taken from approved sources like the lecture examples may be included in the coursework provided these sources are **properly referenced**.
- Failure to reference work that has been obtained from other sources or to copy the words and/or code of another student is plagiarism and, if detected, this will be reported to the School's Discipline Committee. If a student is found guilty of plagiarism, the penalty could involve voiding the course.
- Students must **never** give hard or soft copies of their coursework reports or code to another student. Students must **always refuse** any request from another student for a copy of their report and/or code.
- Sharing a coursework report and/or code with another student is **collusion**, and if detected, this will be reported to the School's Discipline Committee. If found guilty of collusion, the penalty could involve voiding the course.

Plagiarism will be treated extremely seriously as an act of academic misconduct which will result in appropriate student discipline. All students should familiarise themselves with the university policies around plagiarism which can be found here: https://www.hw.ac.uk/students/studies/examinations/plagiarism.htm