**COMP 3710 Artificial Intelligence**

Winter 2019

Term test II

Student Name: Student Number:

1. CNF, Horn clauses, and resolution refutation.
	1. (2 marks) Convert the next 5 rules and 2 observations to Horn clauses. Note. You need to use proper propositional symbols. E.g., *Hair*, *EatMeat*, …

Rules:

* If *X* has hair, then *X* is a mammal.
* If *X* gives milk, then *X* is a mammal.
* If *X* is a mammal and *X* eats meat, then *X* is a carnivore.
* If *X* has feathers, then *X* is a bird.
* If *X* flies and lays eggs, then *X* is a bird

Observations:

* *Stretch* has hair*.*
* *Stretch* eat meat.

*Hair -> Mammal*

*Milk -> Mammal*

*Mammal ∧ EatMeat -> Carnivore*

*Feather -> Bird*

*Fly ∧ LayEgg -> Bird*

*Hair*

*EatMeat*

* 1. (2 marks) Convert the above Horn clauses to a conjunction of disjunction clauses (i.e., CNF form.)

(~Hair ∨ Mammal) ∧ (~Milk ∨ Mammal) ∧ (~Mammal ∨ ~EatMeat ∨ Carnivore) ∧ (~Feather ∨ Bird) ∧ (~Fly ∨ ~LayEgg ∨ Bird) ∧ Hair ∧ EatMeat

* 1. (3 marks) Using resolution refutation, prove or disprove that *Stretch* is a carnivore. You should show one step for the elimination of one symbol. Do not eliminate multiple propositional symbols at once.
* {(~Hair, Mammal), (~Milk, Mammal), (~Mammal, ~EatMeat, Carnivore), (~Feather, Bird), (~Fly, ~LayEgg, Bird), Hair, EatMeat, ~Carnivore}
* {Mammal, (~Milk, Mammal), (~Mammal, ~EatMeat, Carnivore), (~Feather, Bird), (~Fly, ~LayEgg, Bird), EatMeat, ~Carnivore}
* {(~Milk, Mammal), (~EatMeat, Carnivore), (~Feather, Bird), (~Fly, ~LayEgg, Bird), EatMeat, ~Carnivore}
* {(~Milk, Mammal), Carnivore, (~Feather, Bird), (~Fly, ~LayEgg, Bird), ~Carnivore}
* Falsesum because Carnivore, ~Carnivore
* Therefore Carnivore
	1. (3 marks) Using backward chaining, prove or disprove that *Stretch* is a carnivore. You should use the following table one step by one step as you did in a previous assignment.

|  |  |  |
| --- | --- | --- |
| Facts | The goals to check | Matching rules |
| Hair, EatMeat | Carnivore(?) | Carnivore - [Mammal, EatMeat] |
| Hair, EatMeat | Mammal(?) && EatMeat(?) | Mammal - [Hair], [Milk] |
| Hair, EatMeat | (Hair(?) || Milk(?)) && EatMeat(?) | Hair - TRUE |
| Hair, EatMeat, Mammal | (TRUE || Milk(?)) && EatMeat(?) |  |
| Hair, EatMeat, Mammal | TRUE && EatMeat(?) | EatMeat - TRUE |
| Hair, EatMeat, Mammal | TRUE && TRUE |  |
| Hair, EatMeat, Mammal | TRUE |  |

1. (3 marks) Here is the training data set for class grades. Classify (3, 3, 3) by using the 3-nearest neighbor algorithm. You should show how to decide the grade level for the query.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COMP 4620 | COMP 3540 | COMP 3710 | Grade level |  |  |
| 4 | 3 | 4 | Excellent |  | Sqrt(1^2 + 0^2 + 1^2) |
|  3 | 4 | 3 | Good |  | Sqrt(0^2 + 1^2 + 0^2) |
| 3 | 4 | 4 | Excellent |  | Sqrt(0^2 + 1^2 + 1^2) |
| 2 | 3 | 3 | Okay |  | Sqrt(1^2 + 0^2 + 0^2) |
| 3 | 3 | 3 | Good |  | Sqrt(0^2 + 0^2 + 0^2) |
| 2 | 3 | 2 | Okay |  | Sqrt(1^2 + 0^2 + 1^2) |

Good, Okay, Good => Good

1. (3 marks) Compute the information gain for *Temperature*. You do not have to compute the logarithms. Here are two related formulas.

Information Gain = 1 – ∑ (the weighted entropies)

Entropy = – *p*0 × (log2 *p*0) – *p*1 × (log2 *p*1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play** |
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |

Entropy for Hot = –1/3 log2 1/3 – 2/3 log2 2/3

Entropy for Mild = –1/2 log2 1/2 – 1/2 log2 1/2

Entropy for Cool = –3/4 log2 3/4 – 1/4 log2 1/4

Information gain = 1 – (3/9 (–1/3 log2 1/3 – 2/3 log2 2/3) + (2/9 (–1/2 log2 1/2 – 1/2 log2 1/2)) +

(4/9 (–3/4 log2 3/4 – 1/4 log2 1/4)))

1. (3 marks) Here is a perceptron with two input links and one output link. A step function with the threshold *t* is used. Note *w* = *w* + α \* *x* \* *e*
* Input values: *x*1 = 1, *x*2 = 1
* Expected output: 0
* Weights: *w*1 = –0.2, *w*2 = 0.4
* Learning rate: α = 0.1
* Threshold: *t* = 0.1

Decide the new weights.

Output = Y(1 \* -0.2 + 1 \* 0.4) = Y(0.2) = 1 because 0.2 >= threshold

Error = Expected output – Output = 0 – 1 = -1

w1 = w1 + α \* x1 \* Error = -0.2 + 0.1 \* 1 \* (-1) = -0.3

w2 = 0.4 + 0.1 \* 1 \* (-1) = 0.3