Assignment 3

1. Game Tree with Three players

a) Game Properties

Three Players: A, B, and C

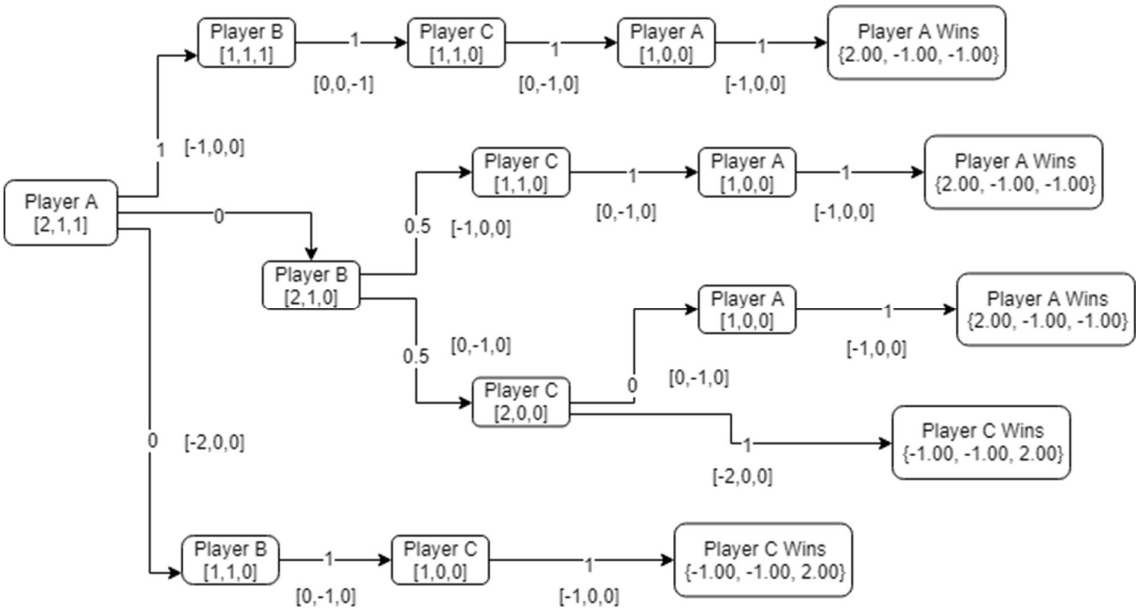
Zero-sum: A wins and B, C lose or B wins and A, C lose or C wins and A, B lose

Deterministic: Perfect Knowledge there is no random aspect to the game

Search Horizon

b) Since there are now three players in the game a good way to represent each leaf node is an interval with three fields. Each field would represented a player's money if they chose that node. The node's interval would look like this [A,B,C] so for example a leaf node could look like [-1.00,2.00,-1.00].

c) Complete Game Tree



d) Player A should take one stick from the pile of two. That limits player B and C to take one stick from the 3 single piles of sticks. After player B, and C move it will leave one pile with a stick for player A to take.

2. Propositional Logic

The cat will play if and only if it wants attention.

If it is 5pm, the cat wants dinner.

If the cat is meowing, it wants either dinner or attention.

If the cat is not meowing, then it is not 5pm.

The cat is meowing.

Use the following lexicon:

Propositional Symbols:

- s -– The cat is meowing.
- a -– The cat wants attention.
- w -– The cat wants dinner.
- i -– It is 5pm.
- p -– The cat will play.

PL:

1. $a \Leftrightarrow p$
2. $i \Rightarrow w$
3. $s \Rightarrow p \vee s \Rightarrow w$
4. $\neg s \Rightarrow \neg i$
5. s

CNF:

1.	$\neg a \vee p$ or $\neg p \vee a$	Clause 1 of theory
2.	$\neg i \vee w$	Clause 2 of theory
3.	$\neg s \vee p$ or $\neg s \vee w$	Clause 3 of theory
4.	$\neg s \vee \neg i$	Clause 4 of theory
5.	s	Clause 5 of theory
6.	p	3 + 5
7.	a	1 + 6
8.	w	3 + 5
9.	i	4 + 5

3. First-Order Predicate Logic

All pandas are herbivores.

No carnivores eat plants.

Herbivores all eat plants.

Use the following lexicon:

Predicates :-

pand(X) -- X is a panda.

herb (X) -- X is an herbivore.

carn(X) -- X is a carnivore.

plant(X) – X eats plants.

FOPL

1. ∀X (pand(X) ⇒ herb(X))

2. ∀X (carn(X) ⇒ ¬plant(X))

---

3. ∀X (herb(X) ⇒ plant(X))

4. First-Order Predicate Logic

All cats are mammals.

The head of any cat is the head of a mammal.

Use the following lexicon:

Predicates :-

cat(X) -- X is a cat

mammal(X) -- X is a mammal

headOf(H,X) -- H is the head of X.

FOPL

1. ∀X (cat(X) ⇒ mammal(X))

---

2. ∀H∃V (headOf(H,V) ⇒ headOf(H,mammal(V)))

CNF

¬cat(x) ∨ mammal(x)

∀H∃x headOf(H(x), cat(H(x))) ^ headOf(H(x), mammal(H(x)))

5.

All horses are prey animals.

Prey animals all run fast.

No meat eaters run fast.

Prove:

No horses are meat eaters.

Use the following lexicon:

Predicates :-

h(X) – X is a horse.

p(X) – X is a prey animal.

f(X) – X runs fast.

m(X) – X eats meat.

FOPL

$\forall h(x) \Rightarrow p(x)$

$p(x) \Rightarrow f(x)$

$m(x) \Rightarrow \neg f(x)$

$\neg h(x) \Rightarrow m(x)$ $1 + 3$

CNF

$\neg h(x) \vee p(x)$

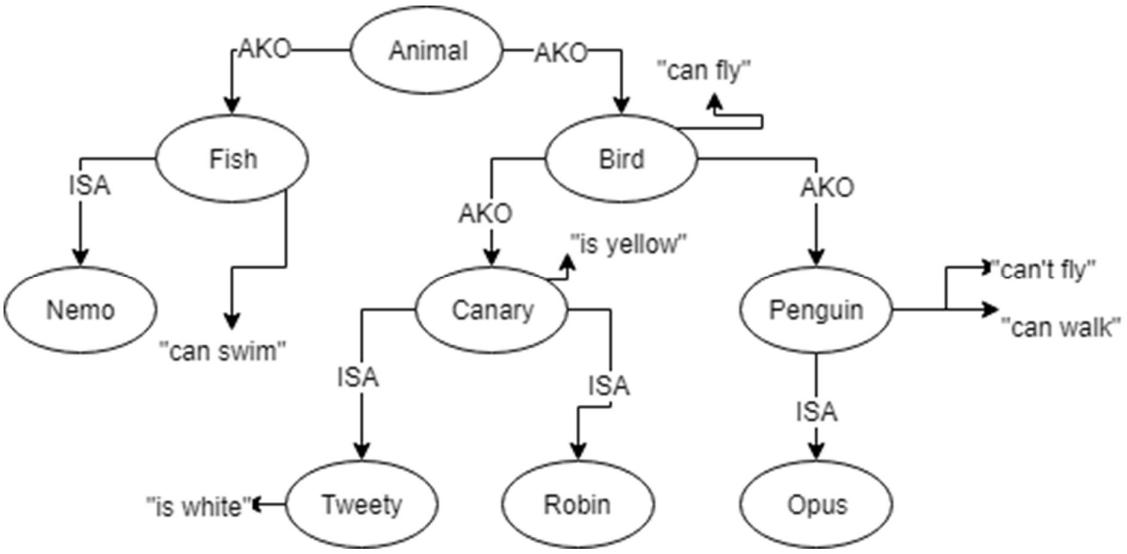
$\neg p(x) \vee f(x)$

$\neg m(x) \vee \neg f(x)$

$\neg((h(horse) \vee \neg m(horse)) \wedge (\neg h(horse) \vee m(horse))$

6.

a) (5 pts.) Draw this taxonomy as a graph, with ``animal" at the root, and label the edges with AKO or ISA, whichever is appropriate.



README

Getting Started

The instructions will get a running copy of the program up for testing.

Prerequisites

Make sure your java is updated to the most recent version on your computer. Make sure all files are present in the scripts and input folders.

Running The Program

When running the program, there are multiple commands. In order to run the program make sure your compile both the Main.java and Treenode.java files. Then run the Main file to launch the program.

Navigate to the following folder path before running commands
jmtimper_Assignment_3/jmtimper_CodingPart/scripts

When running the program the following commands are:

- read <filepath> - reads file of commands"
- edge(<sourceNode>, <linkType>, <destinationNode>) - adds edge with linkType
- rel(<sourceNode>, <destinationNode>) - prints relationship between nodes

- property(<node>, <type>, <value>) - adds property to node
- get(<node>, <type>) - returns property value based on type
- getAll(<node>) - returns all properties including inherited properties
- info(<node>) - returns all information of the node

Test Setup

There are two ways to set up the example tree.

First way:

Enter the following commands:

edge(bird, isa, animal)

edge(fish, isa, animal)

edge(penguin, ako, bird)

edge(canary, ako, bird)

edge(Opus, isa, penguin)

edge(Tweety, isa, canary)

edge(Robin, isa, canary)

edge(Nemo, isa, fish)

Second Way

There are two test files found within the inputs folder. By running the command-animals.txt file, the program will set up the example tree and will test all the functionality in the program. By running the setup-animals.txt file, the program will set up the example tree but won't test the functionality.

Authors

- Jeremy Timperio - jmtimper