HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

T-622-ARTI, Introduction to Artificial Intelligence
Spring 2009

Problem Set 2

Worth 5% of final course grade
Due by 23:59 on Monday the 16th of March

Student Name: ____

1. (28%) Inference with Propositional Logic

You are a sentry system in the secret island hideout of a mad scientist. Your precious Knowledge Base contains the following sentences:

[A] $\texttt{RedLight} \Rightarrow \texttt{PowerOut}$
[B] $\texttt{PowerOut} \Rightarrow \neg\ \texttt{CaveSecure} \vee \texttt{TunnelOpen}$
[C] $\texttt{AreaSecure} \Rightarrow \texttt{CaveSecure} \wedge \neg\ \texttt{TunnelOpen}$
[D] $\neg\ \texttt{AreaSecure} \wedge \texttt{PowerOut} \Rightarrow (\texttt{RedLight} \Rightarrow \texttt{Malfunction})$
[E] $\texttt{BlueLight} \wedge \texttt{Intruder} \Rightarrow \neg\ \texttt{Malfunction}$

The scientist just saw a few lights turn on in the security panel and is panicking. He asks you if you think there is an intruder in the hideout. You sense that the following lights are on:

[F] $\texttt{BlueLight}$
[G] $\texttt{RedLight}$
[H] $\texttt{GreenLight}$

Use inference to determine whether an intruder is present or not. You may use the rules and equivalences shown on the attached formula page. Each time you add a new sentence to your Knowledge Base, **indicate what sentences, equivalences and/or rules you are using** (*note:* number of lines below does not indicate length of solution).

[I] ______________________________ from ____________

[J] ______________________________ from ____________

[K] ______________________________ from ____________

[L] ______________________________ from ____________

[M] ______________________________ from ____________

[N] ______________________________ from ____________

[O] ______________________________ from ____________

[P] ______________________________ from ____________

2. (21%) First Order Logic

Write first order logic statements that express the following:

a. There is a book on my shelf.

b. No book is perfect.

c. All books contain some truth.

3. (30%) Knowledge Representation and Reasoning (with PowerLoom)

a. Work through the attached “Doing Knowledge Representation and Reasoning with PowerLoom” supplement (follow the provided link to download the system).

b. Add more family relations and then insert your own family tree with assertions (if you’re uncomfortable sharing your own family information, you may use a fictional family instead). **Submit your PowerLoom module file (*.PLM) as part of this problem set** (you can zip the files together if you want).

c. How do you ask the system to show you your male cousins on your mother’s side – if they exist?

4. (21%) Planning

A character in a role playing game is played by an autonomous agent. The agent starts off in its own house but wants to get outside and embark on a quest to slay a monster and collect a treasure. The agent engages a hierarchical planner that decomposes the quest into several sub-plans, the first of which is simply to get out of the house with the right equipment. The agent is now trying to solve this first sub-plan and has come up with the following extended STRIPS style formulation of the problem:

Action(*Go(x, y)* ***)***
Precond: $\neg blocked(y) \wedge at(agent, x)$
Effect: $at(agent, y) \wedge \neg at(agent, x)$

Action(*Open(d)* ***)***
Precond: $at(agent, hallway) \wedge \neg locked(d) \wedge connects(d, p) \wedge blocked(p)$
Effect: $\neg blocked(p)$

Action(*Unlock(d, k)* ***)***
Precond: $locked(d) \wedge has(agent, k) \wedge keyfits(k, d)$
Effect: $\neg locked(d)$

Action(*Take(o)* ***)***
Precond: $at(agent, x) \wedge at(o, x) \wedge \neg has(agent, o)$
Effect: $has(agent, o) \wedge \neg at(o, x)$

Initial State:
$at(agent, hallway) \wedge at(key, hallway) \wedge at(sword, room) \wedge blocked(outside) \wedge \neg blocked(hallway) \wedge \neg blocked(room) \wedge connects(door, outside) \wedge locked(door) \wedge keyfits(key, door)$

Goal:
$at(agent, outside) \wedge has(agent, sword)$

a. Is the following sequence of actions a solution to this planning problem? Why / why not?

Unlock(door, key)
Open(door)
Go(hallway, room)
Take(sword)
Go(room, outside)

Answer:

b. What is the state after executing the following actions?

Take(key)
Go(hallway, room)
Take(sword)
Go(room, hallway)
Unlock(door, key)

Answer:

c. Perform the initial steps in planning through **backward chaining** by **regressing the final goal** above through at least two actions. Describe the states from which these actions are applied.

Formula Page

Logical Equivalence:

1. $(a \wedge b) \equiv (b \wedge a)$
2. $(a \vee b) \equiv (b \vee a)$
3. $((a \wedge b) \wedge c) \equiv (a \wedge (b \wedge c))$
4. $((a \vee b) \vee c) \equiv (a \vee (b \vee c))$
5. $\neg(\neg a) \equiv a$
6. $(a \Rightarrow b) \equiv (\neg b \Rightarrow \neg a)$
7. $(a \Rightarrow b) \equiv (\neg a \vee b)$
8. $(a \Leftrightarrow b) \equiv ((a \Rightarrow b) \wedge (b \Rightarrow a))$
9. $\neg(a \wedge b) \equiv (\neg a \vee \neg b)$
10. $\neg(a \vee b) \equiv (\neg a \wedge \neg b)$
11. $(a \wedge (b \vee c)) \equiv ((a \wedge b) \vee (a \wedge c))$
12. $(a \vee (b \wedge c)) \equiv ((a \vee b) \wedge (a \vee c))$

Sound Inference Rules:

13. From $\{ (a \Rightarrow b), a\}$ infer b (Modus Ponens)
14. From $\{ (a \Rightarrow b), \neg b\}$ infer $\neg a$ (Modus Tolens)
15. From $\{ a, b \}$ infer $(a \wedge b)$
16. From $\{ (a \wedge b), . \}$ infer a
17. From $\{ (a \wedge b), . \}$ infer b

Doing Knowledge Representation and Reasoning with PowerLoom.

You get PowerLoom here: http://www.isi.edu/isd/LOOM/PowerLoom/

Create a new, empty, module to work in and specify representation language syntax:

```
(defmodule "PL-USER/FAMILY")
(in-module "FAMILY")
(reset-features)
(in-dialect KIF)
```

Save your module to disk (show up in your powerloom folder or the "kbs" subfolder), and loading it back later:

```
(save-module "FAMILY" "FAMILY.PLM")

(load "FAMILY.PLM")
(in-module "FAMILY")  ;;; If not already in this module
```

Defining a basic type/class predicate (a unary relation) called a "concept" in Powerloom:

```
(defconcept Person(?p))
(defconcept Male(?p))
(defconcept Female(?p))
```

Basic TELLing and ASKing in Powerloom:

```
(assert (Male John))
(assert (Female Mary))

(ask (Male John))
(ask (Female Mary))
```

Adding FOL axioms:

```
(assert (forall (?x) (=> (Male ?x) (Person ?x))))
;; Being a male implies you are a person
;; do the same for females
```

Asking for possible substitutions:

```
(retrieve (Person ?p))    ;;; Returns one possible substitutions for ?p if it exists
(retrieve all (Person ?p)) ;;; Returns all possible substitutions for ?p
```

The Open-World semantics of Powerloom:

```
(ask (Male Mary))  ;;; Should be unknown since it wouldn't conflict with the KB
(assert (forall (?p) (<=> (Male ?p) (not (Female ?p)))))
;; Being a male implies you are not a female
;; do the same for females
(ask (Male Mary))
```

Defining a regular relation predicate in Powerloom:

```
(defrelation BrotherOf ((?p1 Male) (?p2 Person)))
(assert (BrotherOf John Mary))
(assert (Person Olaf))
(assert (BrotherOf Olaf Mary))

(retrieve all (BrotherOf ?x Mary))
(ask (Male Olaf))

;; Create a new predicate called ParentOf
```

Defining a regular function in Powerloom:

```
(deffunction GetFather ((?p1 Person)) :-> (?p2 Person)))

(assert (= (GetFather Mary) Zod))
```

A new axiom that uses a function and equivalence:

```
(assert (<=> (= (GetFather ?c) ?f) (and (Male ?f) (ParentOf ?f ?c))))

;; Answer: Is Zod Male?  Is he Mary‘s parent?
```

Now you are on your own.... add more family relations like SisterOf, GrandmotherOf, AreSiblings, UncleOf, SonOf, AuntOf, DaughterOf, ChildOf, ... and try answering questions like „Is X a sibling of Y?“, „who are X‘s grandmothers?“, „who are X‘s uncles?“, „Does X‘s mother‘s mother have a male child?“...

Helpful commands:

```
(all-facts-of Zod) ;;; prints out all known facts about Zod
(help assert) ;;; prints out help text about “assert”
(demo) ;;; leads you through various PowerLoom demonstrations
```