

# Exercises for Artificial Intelligence

Dr Sean B Holden, 2010-20

## 1 Knowledge representation and reasoning

1. There were in fact *two* queries suggested in the notes for obtaining a sequence of actions. The details for

$$\exists a \exists s . \text{Sequence}(a, s_0, s) \wedge \text{Goal}(s)$$

were provided, but earlier in the notes the format

$$\exists \text{actionList} . \text{Goal}(\dots \text{actionList} \dots)$$

was suggested. Explain how this alternative form of query might be made to work.

2. Making correct use of the situation calculus, write the sentences in FOL required to implement the Shoot action in Wumpus World. Write further sentences in FOL to allow movement and change of orientation.

3. Download and install a copy of *Prover9* from [www.cs.unm.edu/~mccune/prover9/](http://www.cs.unm.edu/~mccune/prover9/). (If you're Linux-based then you'll probably find it's already packaged. Under OS X it appears to be available through HomeBrew. As for Windows, I have no idea. It's just too awful to contemplate!)

Referring to exam question 2003, paper 9, question 8 assume that initially both owner and cat are in the living room. The cat can make its owner move to the kitchen by going to its food bowl in the kitchen and meowing. It can then of course return to the living room and scratch something valuable.

Implement sufficient knowledge in the situation calculus to allow an action sequence to be derived allowing the cat to achieve this, and use Prover9 to derive such an action sequence.

In order to do this you need to know how to extract an answer from the theorem prover. Taking an easy example:

```
formulas(assumptions).
```

```
wife(x,y) <-> (female(x) & married(x,y)).  
female(Violet).  
married(Violet,Bill).
```

```
endoflist.
```

```
formulas(goals).
```

```
exists x wife(Violet,x).
```

```
endoflist.
```

Extracting the value of  $x$  requires two things: we need to move the goal into the assumptions (remember how in *Logic and Proof* you converted  $\neg(A \rightarrow B)$  to  $A \wedge \neg B$  when negating and converting to clauses) and we need to add a command to the knowledge base to get:

```
formulas(assumptions).  
  
wife(x,y) <-> (female(x) & married(x,y)).  
female(Violet).  
married(Violet,Bill).  
  
-wife(Violet,x) # answer(x).  
  
endoflist.  
  
formulas(goals).  
endoflist.
```

Here, the addition of `# answer(x)` causes the prover to output the value of `x` as part of the solution.