Artificial Intelligence

Intro to Prolog

So, what it Prolog?

- Programmation en Logique
- Yep, you guessed it, it was developed by a French guy, *Alain Colmerauer*, in the early '70s
- It's a declarative language: say *what* you want done, not *how* to do it

Programming in Prolog

- Declare explicit *facts* that state relations between objects
- Declare *rules* that allow the *inference* of new facts
- Ask *questions* about object relations or object properties

Relations

- Relations are important in Prolog: a program specifies relations amongst objects
- “Alan owns a typewriter” defines the ownership relation between two objects: Alan and the typewriter

More on relations

Relations can be expressed as rules if they are more complicated:

```
Two people are sisters if
   they are both female and
   they have the same parents
```

Making the relation clear

A better definition of sisterhood:

```
X and Y are sisters if
    X and Y are both female and
    they have the same father and
    they have the same mother and
    X is not the same as Y
```

Facts

- They define relations between objects or object properties
- The earlier example regarding Alan would be written as:

```
owns(alan, typewriter).
```

More on facts

- They are first order predicates
- The number of arguments gives the predicate arity (the predicate we defined earlier has arity 2)
- When talking about facts we often use the following notation: `fact/arity` (e.g. `owns/2`)

A tad more on facts

- The order of arguments once defined must be obeyed
- All the facts in a Prolog program define the *knowledge base* of that program (we'll see later that rules are also part of the knowledge base)

Queries

- Once we have a knowledge base we can query it
- Queries have a truth value in the context of the facts in the knowledge base
- Queries can be seen as yes/no questions (note: SWI-Prolog will answer with `true` or `false` instead of `yes` and `no`)

Variables

- Suppose we want to want to know what object is it that Alan owns
- We could phrase our question as:

```
Is there an object X, that Alan
owns?
```

- X is a variable

More on variables

- A variable without a value is called *unbound*
- Once a variable gets bound it can’t be rebound
- When we don’t care about the value of a variable we can use _
- Variables are bound by *unification* (we’ll talk about it later)

Rules

- Rules have the following form:

```
S :- S1, S2, …, Sn.
```

- S is the *head* of the rule, an S1, S2, ..., Sn is the *body*
- :- can be read as *if* and is sometimes called the *neck* of the rule
- , can be read as *and*

More on rules

- The rule body is a *conjunction* of *goals*
- Using rules we can deduce new relations using existing ones
- Rules are also called *clauses*

A rule example

Remember the sisterhood rule we defined earlier?

```
X and Y are sisters if
   X and Y are both female and
   they have the same father and
   they have the same mother and
   X is not the same as Y
```

A rule example (II)

We could write it as:

```
sisters(X, Y) :-
    female(X), female(Y),
    father(A, X), father(A, Y),
    mother(B, X), mother(B, Y),
    X \== Y.
```

The inner workings of Prolog

- When trying to *satisfy* a clause Prolog tries to satisfy each goal in the clause’s body left to right
- When satisfying a goal fails, Prolog returns to the previous goal and tries *re-satisfying* that after clearing all bindings for the currently bound variables
- This is known as *backtracking*

The syntax of Prolog

- Prolog has two types of terms: *basic* and *compound* (we'll talk about these later)
- Basic terms are *constants* and *variables*
- Constants are: *numbers* and *atoms*

Syntax of atoms

- Atoms are character sequences that begin with a *lowercase letter* and can contain any mix of alphanumeric characters and the underscore “_”
- When an atom must begin with an uppercase letter or it must contain special characters we can escape it by placing it in single quotes

Syntax of variables

- Variables must begin with an upper case letter and they can contain any mix of alphanumeric characters and the underscore "_"

Homework

You’ll find a PDF file with a list of problems on my page:

http://muscar.github.com/teaching/2011/spring/ai.html

They are due next lab.