KNOWLEDGE REPRESENTATION, REASONING AND DECLARATIVE PROBLEM SOLVING

By Jicheng Zhao
based on the notes of others

Class 11: Inheritance Hierarchy and Prioritized default

Inheritance Hierarchies

Ansprolog representations of normative statements discussed in the previous classes do not encode the properties of normative statements such as ‘Normally elements of class C have the property P’ into a hierarchy.

Example 1 *Given the following program.*

$$\begin{aligned} fly(X) &\leftarrow bird(X), \textbf{not}\ penguin(X). \\ bird(X) &\leftarrow penguin(X). \end{aligned}$$

The relationships can be represented graphically.

There are three categories of edges in the graph.

1. The property at the head of the arrow holds for the class at the tail;
2. The property at the head of the arrow does not hold for the class at the tail;

3. Refers to an 'isa' relationship. The class at the tail of the arrow is a type of the class at the head of the arrow.

The property of a base class is inherited by the sub class through the isa relationship. However, the property bound to a more specific class overrides the property bound from a generic class. In other words, a property of *penguin* will override a property of *bird*, which is a superset of *penguin*. Hence, *penguins* do not normally fly, even though *birds* fly. The logic program to this problem is:

$$\begin{aligned} fly(X) &\leftarrow bird(X), \mathbf{not}\ ab(X). \\ havewings(X) &\leftarrow bird(X), \mathbf{not}\ ab(X). \\ bird(X) &\leftarrow penguin(X). \\ ab(X) &\leftarrow penguin(X). \end{aligned}$$

Another way to encode the birds example in ansprolog would be:

$$fly(X) \leftarrow bird(X), \mathbf{not}\ \neg fly(X).$$

$$
\begin{aligned}
havewings(X) &\leftarrow bird(X), \mathbf{not}\ \neg havewings(X).\\
bird(X) &\leftarrow penguin(X).\\
\neg fly(X) &\leftarrow penguin(X).
\end{aligned}
$$

Dealing with Contradictions

Example 2 *The example illustrates a case where a contradiction occurs.*

- *Nixon is a Republican;*
- *Nixon is a Quaker;*
- *Quakers normally are pacifists;*
- *Republicans normally are not pacifists.*

The goal: to reason if Nixon is a pacifist. The intuitive solution would be to arrive at no decision. But Ansprolog reasoning paves ways for two possible conclusions.

1. No decision can be arrived at. The answer set returns all false. or
2. There are two world views (answer sets). Nixon is a pacifist in one answer set and he isn't in the other one.

To arrive at a solution that we still unknown whether $Nixon$ is a pacifist or not, we would encode it as:

$$\begin{aligned} quaker(nixon). & \\ republican(nixon). & \\ pacifist(X) & \leftarrow quaker(X), \textbf{not } ab1(X). \\ \neg pacifist(X) & \leftarrow quaker(X), \textbf{not } ab2(X). \\ ab1(X) & \leftarrow republican(X). \\ ab2(X) & \leftarrow quaker(X). \end{aligned}$$

The following encoding gives us two answer sets, one in which Nixon is a pacifist and one in which Nixon is not a pacifist.

$$\begin{aligned} quaker(nixon). & \\ republican(nixon). & \\ pacifist(X) & \leftarrow quaker(X), \textbf{not } \neg pacifist(X). \\ \neg pacifist(X) & \leftarrow republic(X), \textbf{not } pacifist(X). \end{aligned}$$

(1)

Example 3 *Consider the case where there is a property, f, which holds for both pacifists and non-pacifists. How to present the problem such that Nixon has the property f.*

In the first case, there would be no reasoning about the property f, since we were not able to arrive at any conclusion.
Whereas in the latter case, we will be able to reason that the property f holds for both Quakers and Republicans, and hence to Nixon, by the class inheritance hierarchy. This reasoning about property f is made possible in both the world views. We can just add the following two rules to accomplish this:

$$f(X) \leftarrow pacifist(X).$$
$$f(X) \leftarrow \neg pacifist(X).$$

(2)

The Ansprolog programs that go about each of the reasoning methods mentioned above can be considered as following the skeptical or the credulous reasoning approaches.

One more example

Consider a legal reasoning system, with the following rules.

Example 4 *Assume a person wants to find out if his security interest in a certain ship is perfected. He (John) currently has possession of the ship. According to the Uniform Commercial Code (ucc, §9-305) a security interest in goods may be perfected by taking possession of the collateral. However, there is a federal law called the Ship Mortgage Act (sma) according to which a security interest in a ship may only be perfected by filing a financing statement. Such a statement has not been filled. Now the question is whether the ucc or the sma takes precedence in this case. There are two known legal principles for resolving conflicts of this kind. The principle of Lex Posterior gives precedence to newer laws. In our case the ucc is newer than the sma. On the other hand, the principle of Lex Superior gives precedence to laws supported by the higher authority. In our case the sma has higher authority since it is federal law.*

The above set of rules has to be encoded in Ansprolog. The intuitive reasoning in this case would lead to a result where no decision can be made with regards to the possession of the ship. A possible way to arrive at a conclusion would be to arrive at some kind of precedence between Lex posterior and Lex superior rules. Another way suggested would be to give way to specificity of objects, considering ship to be more specific than goods, and so, considering the sma to hold for the current situation of John's ship.
An initial set of rules and facts from the legal reasoning example would be:

$$
\begin{aligned}
perfected &\leftarrow possessed. \\
\neg perfected &\leftarrow \neg filled. \\
precedence(X,Y) &\leftarrow newer(X,Y). \\
precedence(X,Y) &\leftarrow federal(X), state(Y). \\
\neg filled.
\end{aligned}
$$

$$
\begin{aligned}
possessed.\\
federal(sma).\\
state(ucc).\\
newer(ucc, sma).
\end{aligned}
$$

A possible way to encode the conditions would be,

$$
\begin{aligned}
perfected &\leftarrow possessed, prefer(ucc, sma).\\
\neg perfected &\leftarrow \neg filled, prefer(sma, ucc).
\end{aligned}
$$

But this enumeration would be time consuming and does not help in generalizing the rules for any law.
We could also encode it as,

$$
\begin{aligned}
perfected &\leftarrow possessed, preferred\_over\_others(ucc).\\
\neg perfected &\leftarrow \neg filled, preferred\_over\_others(sma).\\
preferred\_over\_others(X) &\leftarrow \textbf{not } \neg prefer(Y, X).\\
\neg prefer(Y, X) &\leftarrow prefer(X, Y), Y \neq X.
\end{aligned}
$$

But this method fails if we have a rule $L3$ which is preferred over ucc, but whose conditions are not satisfied. In which case, we should reason with ucc, but because of $L3$ being preferred over ucc ($preferred\_over\_others(ucc)$ would fail, even thought ucc is preferred now that $L3$ does not hold, considering $L3$ to be the only law that is preferred over ucc).

Solution 3:

$$\begin{aligned}
holds(prefect, ucc) &\leftarrow possessed. \\
\neg holds(prefect, sma) &\leftarrow \neg filled. \\
precedence(X, Y) &\leftarrow newer(X, Y). \\
precedence(X, Y) &\leftarrow federal(X), state(Y). \\
\neg filled. & \\
possessed. & \\
federal(sma). & \\
state(ucc). &
\end{aligned}$$

$$
\begin{aligned}
newer(ucc, sma). & \\
conclude(prefect) &\leftarrow holds(prefect, X), \mathbf{not}\ override(prefect, Y, X). \\
\neg conclude(prefect) &\leftarrow \neg holds(prefect, X), \mathbf{not}\ override\_neg(prefect, Y, X). \\
override(prefect, Y, X) &\leftarrow precedence(Y, X), \neg holds(prefect, Y). \\
override\_neg(prefect, X) &\leftarrow precedence(Y, X), holds(prefect, Y).
\end{aligned}
$$

Prioritized default

It is one approach where normative statement, exceptions, preferences, etc. are represented as facts, and there are general purpose rules that take this facts and reason whit them.
Each of the laws has to be associated with their names. One way to label the rules for the laws would be to encode the rules and the labels as a fact in Ansprolog. For example, the ucc law would be encoded as

$$default(ucc, perfected, possessed) \leftarrow .$$

Where the format is default (*label, head, [body]*) $\leftarrow$., the body being a list of atoms.
The legal scenario would thus be encoded as,

$$default(ucc, perfected, [possessed]) \leftarrow .$$
$$default(sma, \neg perfected, [\neg filled]) \leftarrow .$$
$$default(lex\_posterior, prefer(D1, D2), [more\_recent(D1, D2)]) \leftarrow .$$
$$default(lex\_superior, prefer(D1, D2), [federal(D1), state(D2)]) \leftarrow .$$
$$\neg filled \leftarrow .$$
$$possessed \leftarrow .$$
$$more\_recent(ucc, sma) \leftarrow .$$
$$federal(sma) \leftarrow .$$
$$state(ucc) \leftarrow .$$

A suggested improvement to the above rules would be to include the idea that preference will be considered only for laws that contradict in their ruling. Hence,

$$default(ucc, perfected, [possessed, contradict(ucc, X), applicable(X), preferred(ucc, X)]) \leftarrow .$$
$$contradict(X, Y) \leftarrow default(X, U, \_), default(Y, \neg U, \_), \mathbf{not}\ \neg contradict(X, Y).$$

$$\neg contradict(X, Y) \leftarrow \mathbf{not}\ contradict(X, Y).$$

$$applicable(ucc) \leftarrow possession.$$

$$applicable(sma) \leftarrow \neg filled.$$