Encode *birds fly*

\[
\text{fly}(X) :- \text{bird}(X).
\]

to allow for exceptions.
Encode *birds fly*

\[
\text{fly}(X) :- \text{bird}(X).
\]

\%
\[
\frac{\text{bird}(X)}{\text{fly}(X)}
\]

to allow for exceptions.

*Default rule* (R. Reiter)

\[
\frac{\text{bird}(X) : \text{fly}(X)}{\text{fly}(X)}
\]
Encode \textit{birds fly}

\[
\text{fly}(X) :- \text{bird}(X).
\]

\[
\begin{array}{l}
\frac{\text{bird}(X) : \text{fly}(X)}{\text{fly}(X)}
\end{array}
\]

to allow for exceptions.

\textit{Default rule} (R. Reiter)

In general, applied to $KB$ says:

conclude $c$ if $KB \models p$ and $j$ is $KB$-consistent

$KB, j \not\models false$
Rules and defaults

Encode *birds fly*

\[
\text{fly}(X) :- \text{bird}(X). \quad \%
\]

\[
\frac{\text{bird}(X) \quad \text{fly}(X)}{\text{fly}(X)}
\]

To allow for exceptions.

*Default rule* (R. Reiter)

In general,

\[
\text{prerequisite } p : \text{justification } j \quad \rightarrow \quad \text{conclusion } c
\]

Applied to \( KB \) says:

\[
\text{conclude } c \quad \text{if} \quad KB \models p \text{ and } j \text{ is } KB\text{-consistent}
\quad \text{and } \quad KB, j \not\models \text{false}
\quad j \text{ is true in some model of } KB
Let $KB$ be

\begin{align*}
\text{bird(robin)}. \\
\text{bird(penguin)}. \\
\text{false :- fly(penguin)}. \\
\text{fly(bee)}. \\
\end{align*}

Conclude:
Birds and bees

(\(\star\)) \[ \frac{\text{bird}(X) : \text{fly}(X)}{\text{fly}(X)} \]

Let \(KB\) be

- \(\text{bird(robining)}\).
- \(\text{bird(penguin)}\).
- \(\text{false} :- \text{fly(penguin)}\).
- \(\text{fly(bee)}\).

Conclude:

- \(\text{fly(robining)}\) by default rule (\(\star\))

but \(\text{not fly(penguin)}\).
Birds and bees

\[(\star) \quad \frac{\text{bird}(X) : \text{fly}(X)}{\text{fly}(X)}\]

Let $KB$ be

\begin{align*}
\text{bird}(\text{robin}). \\
\text{bird}(\text{penguin}). \\
\text{false} :- \text{fly}(\text{penguin}). \\
\text{fly}(\text{bee}).
\end{align*}

Conclude:

\begin{align*}
\text{fly}(\text{robin}) & \quad \text{by default rule (\star)} \\
\text{but not fly}(\text{penguin}). \\
\text{An explanation of fly}(\text{bee}) \text{ using (\star) is} \\
\text{bird}(\text{bee})
\end{align*}
Birds and bees

(⋆) \[
\frac{\text{bird}(X): \text{fly}(X)}{\text{fly}(X)}
\]

Let \( KB \) be

\[
\begin{align*}
\text{bird}(\text{robin}). \\
\text{bird}(\text{penguin}). \\
\text{false} : - \, \text{fly}(\text{penguin}). \\
\text{fly}(\text{bee}).
\end{align*}
\]

Conclude:

\[
\begin{align*}
\text{fly}(\text{robin}) \quad \text{by default rule (⋆)}
\end{align*}
\]

but \textit{not} \text{fly}(\text{penguin}).

An explanation of \text{fly}(\text{bee}) using (⋆) is

\[
\begin{align*}
\text{bird}(\text{bee})
\end{align*}
\]

which we can block by adding to \( KB \) the rule

\[
\begin{align*}
\text{false} : - \, \text{bird}(\text{bee}).
\end{align*}
\]
Conflicting defaults

\[
\text{quaker}(X) : \text{pacificist}(X) \quad \text{republican}(X) : \text{hawk}(X)
\]

Let \( KB \) be quaker(nixon). republican(nixon). false :- pacifist(X), hawk(X).

Applying one default to Nixon makes the other inapplicable. \( KB \) has two incompatible extensions, breaking least fixed point (provability model) for Horn clauses.
Non-determinism

Conflicting defaults

\[
\begin{align*}
\text{quaker}(X) : & \text{pacifist}(X) & \quad \text{republican}(X) : & \text{hawk}(X) \\
\text{pacifist}(X) & & \text{hawk}(X)
\end{align*}
\]

Let $KB$ be

\[
\begin{align*}
\text{quaker}(\text{nixon}). \\
\text{republican}(\text{nixon}). \\
\text{false} :- \text{pacifist}(X), \text{hawk}(X).
\end{align*}
\]
Conflicting defaults

\[
\begin{align*}
\text{quaker}(X) &: \text{pacifist}(X) & \text{republican}(X) &: \text{hawk}(X) \\
\text{pacifist}(X) & & \text{hawk}(X)
\end{align*}
\]

Let \( KB \) be

\[
\begin{align*}
\text{quaker}(\text{nixon}). \\
\text{republican}(\text{nixon}). \\
\text{false} & : - \text{pacifist}(X), \text{hawk}(X).
\end{align*}
\]

Applying one default to Nixon makes the other inapplicable.

\( KB \) has two incompatible extensions, breaking

least fixed point (provability model) for Horn clauses.
A default rule is *normal* if its justification is its conclusion

\[
\frac{p}{c}
\]

– infer \( c \) if it is consistent and \( p \) is provable
Normal default rules and inferring negations

A default rule is *normal* if its justification is its conclusion

\[
\frac{p}{c}
\]

- infer \( c \) if it is consistent and \( p \) is provable

*Closed World Assumption*: any unprovable atom \( \varphi \) is false

\[
\text{true} : \neg \varphi
\]

\[
\frac{\neg \varphi}{\neg \varphi}
\]
Normal default rules and inferring negations

A default rule is *normal* if its justification is its conclusion

\[
\frac{p}{c}
\]

– infer \( c \) if it is consistent and \( p \) is provable

*Closed World Assumption:* any unprovable atom \( \varphi \) is false

\[
\text{true} : \neg \varphi
\]

\[
\frac{}{\neg \varphi}
\]

*Negation as failure:* \( \varphi \) is false if attempting to prove \( \varphi \) fails finitely

\[
naf(P) :- (P,!\!,\text{fail}); \text{true}.
\]
Normal default rules and inferring negations

A default rule is *normal* if its justification is its conclusion

\[
\frac{p : c}{c}
\]

- infer \( c \) if it is consistent and \( p \) is provable

*Closed World Assumption:* any unprovable atom \( \varphi \) is false

\[
\text{true} : \neg \varphi \\
\neg \varphi
\]

*Negation as failure:* \( \varphi \) is false if attempting to prove \( \varphi \) fails finitely

\text{naf}(P) :- (P,!,fail); true.

N.B. Checking finite failure can be as hard as the Halting Problem.