In Praise of Impredicativity: A Contribution to the Formalisation of Meta-Programming

François Bry

21 September 2017
1 Introduction

Amalgamation (Bowen and Kowalski):

\[
\text{believes(ann, itRains)} \\
\text{believes(ann, itIsWet} \leftarrow \text{itRains)} \\
\text{believes(bill, X} \leftarrow \text{believes(ann, X))}
\]

In Prolog:

\[
\text{believes(ann, itRains)} \\
\text{believes(ann, cl(itIsWet, itRains))} \\
\text{believes(bill, cl(X, believes(ann, X)))}
\]
More is possible and meaningful:

\[(\text{loves} \land \text{trusts})(\text{ann, bill})\]
\[(\text{loves} \land \text{trusts})(X, Y) \leftarrow \text{loves}(X, Y) \land \text{trust}(X, Y)\]
\[(P1 \land P2)(X, Y) \leftarrow P1(X, Y) \land P2(X, Y)\]

\[(\forall T \; \text{trust}(T) \Rightarrow T)(\text{ann, bill})\]
\[(T \leftarrow \text{trust}(T))(\text{ann, bill})\]
\[(T \leftarrow \text{trust}(T))(X, Y) \leftarrow \text{not (trust}(T) \land \text{not } T(X, Y))\]
Contribution:

1. A discussion of how meta-programming relates to higher-order logics and to impredicativity.

2. A simplification of the syntax and model theory of Ambivalent Logic, a logic proposed for formalising meta-programming.

3. An explanation why Ambivalent Logic’s impredicativity is acceptable.
Formalisations of meta-programming:

1. relying on higher-order logics: “higher-order logic programming”
   (λ-Prolog, Elf and Twelf – Precludes amalgamation)
2. representing formulas by terms
   (“naming schemes” associating a term to each object formula)
3. Ambivalent Logic, a logic lifting the distinction between terms and formulas.
Amalgamation allows

1. variables to range over predicates and formulas,
2. predicates the arguments of which are predicates or formulas, and
3. reflection: every predicate can have any predicate, including itself, and any formulas as argument.

Higher-order logics allow no confusion of orders like

- A unary predicate ranging over all unary predicates including itself.
- A predicate being, or occurring in, an argument of itself.

Such confusions (or “amalgamations”) are widespread in Prolog-style meta-programming.
4 Predicativity and Impredicativity

- A node $n$ of $G$ has property $P$ if and only if its immediate neighbours all have property $P$.

- $y$ is the smallest element of an ordered set $S$ if and only if for all elements $x$ of $S$, $y$ is less than or equal to $x$, and $y$ is in $S$.

These definitions are “impredicative” (Russell): Each definition refers to the property it defines.

Some impredicative definitions are now considered acceptable:

- Inductive definitions.
- Impredicative definitions that characterise elements of clearly apprehensible sets (including inductively defined sets).
Ambivalent Logic (Kalsbeek and Jiang) has “amalgamation” or “impredicativity” built-in.

An Ambivalent Logic language is defined like a First-Order Logic language by

1. the logical symbols consisting of the connectives $\land$, $\lor$, $\rightarrow$, and $\neg$, and of the quantifiers $\forall$ and $\exists$,
2. at least one (and at most finitely many) non-logical symbols each of which is distinct from every logical symbol.
The expressions of an Ambivalent Logic language are inductively defined as follows:

- A non-logical symbol $s$ is an expression.
- If $E$ and $E_1, \ldots, E_n$ with $n \geq 1$ are expressions, then $E(E_1, \ldots, E_n)$ is an expression.
- If $E$ is an expression, then $(\neg E)$ is an expression.
- If $E_1$ and $E_2$ are expressions, then $(E_1 \land E_2)$, $(E_1 \lor E_2)$, $(E_1 \Rightarrow E_2)$ are expressions.
- If $E_1$ is an expression and if $E_2$ is an expression, then $(\forall E_1 E_2)$ and $(\exists E_1 E_2)$ are expressions.

Paradigm “quantification makes variables”

- departs from Kalsbeek’s and Jiang’s proposal,
- is useful in practice: $\text{likes(ann, bill)}$ is easily turned into $\exists X \text{likes(ann, X)}$ or $\forall X \text{likes(X, bill)}$. 
A (consistent) expression defines a “reflexive set” (a set that can have some of their subsets as elements).

The celebrated paradoxes remain but do not compromise Ambivalent Logic more than First-Order Logic.

Barber Paradox:

\[
\text{man(barber)}
\]
\[
(\forall y \ (\text{man}(y) \Rightarrow (\text{shaves(barber, y)} \iff (\neg \text{shaves}(y, y))))))
\]

The barber cannot exist because he would have both to shave and not to shave himself.

Russell’s Paradox:

\[
(\forall x \ (e(x) \iff (\neg x(x))))
\]

Yields the self-contradicting expression \((e(e) \iff (\neg e(e)))\)
Rectified expression: Quantified subexpressions consistently replaced by standard variables (additional symbols).

Rectified atom: Atom possibly containing standard variables.

\( \mathcal{A} \): Set of rectified atoms of an Ambivalent Logic language, that is, expressions like

\[
\begin{align*}
\text{believes}(\text{bill}, \forall X \text{ believes}(\text{ann}, X)) \\
\text{believes}(\text{bill}, \text{believes}(\text{likes}(\text{ann}, \text{bill}))) \\
\text{student}(\text{ann})
\end{align*}
\]

but unlike

\[
\begin{align*}
\forall X \text{ believes}(\text{ann}, X) \\
\exists X \text{ believes}(\text{ann}, X) \\
(\text{student}(\text{ann}) \land \text{student}(\text{bill})) \\
(\text{student}(\text{ann}) \Rightarrow \text{human}(\text{ann})) \\
\text{student}(X)
\end{align*}
\]
Ambivalent Logic’s Model Theory Revisited

\(\sim\): Variant relation extended to quantified expressions:

\[
\forall X \text{ believes}(\text{ann}, X) \sim \forall Y \text{ believes}(\text{ann}, Y)
\]

\[
\text{believes}(\text{bill}, \forall X \text{ believes}(\text{ann}, X)) \sim \text{believes}(\text{bill}, \forall Y \text{ believes}(\text{ann}, Y))
\]

The Herbrand universe of \(\mathcal{L}\) is \(\mathcal{A}/\sim\), the set of equivalence classes of \(\sim\), like for example the class of both

\[
\text{believes}(\text{bill}, \forall X \text{ believes}(\text{ann}, X)) \]

\[
\text{believes}(\text{bill}, \forall Y \text{ believes}(\text{ann}, Y))
\]

or the class \(\{\text{student(ann)}\}\).

An Herbrand interpretation \(I(S)\) of an Ambivalent Logic language \(\mathcal{L}\) is specified as a subset \(S\) of the universe of \(\mathcal{L}\), that is, a set of equivalence classes for \(\sim\).
Satisfaction in an Herbrand interpretation \( I(S) \) of \( \mathcal{L} \) is defined as follows, where

- \( E, E_1, \) and \( E_2 \) denote *rectified* expressions,
- \( A \) denotes a *rectified* atom,
- \( v \) denotes a variable.

\[
\begin{align*}
I(S) &\models A \iff A \in \text{class}(A) \in S \\
I(S) &\models \neg E \iff I(S) \not\models E \\
I(S) &\models (E_1 \land E_2) \iff I(S) \models E_1 \text{ and } I(S) \models E_2 \\
I(S) &\models (E_1 \lor E_2) \iff I(S) \models E_1 \text{ or } I(S) \models E_2 \\
I(S) &\models (E_1 \Rightarrow E_2) \iff \text{if } I(S) \models E_1, \text{ then } I(S) \models E_2 \\
I(S) &\models \exists v \ E \iff I(S) \models E[A/v] \text{ for some } A \\
I(S) &\models \forall v \ E \iff I(S) \models E[A/v] \text{ for all } A
\end{align*}
\]
In an interpretation, the quantified expressions within an atom like
\[
\text{believes}(\text{bill}, \forall X \text{ believes}(\text{ann}, X))
\]
are not interpreted.

Atoms like
\[
\begin{align*}
\text{believes}(\text{bill}, \forall X \text{ believes}(\text{ann}, X)) \\
\text{believes}(\text{bill}, \forall Y \text{ believes}(\text{ann}, Y))
\end{align*}
\]

are identically interpreted because an Herbrand interpretation is specified as a subset \( S \) of \( \mathcal{A}/\sim \) (the Herbrand universe).
8 Conclusion

Perspectives for further work:

- Relation to First-Order Logic (conjecture: Ambivalent Logic is expressible in First-Order Logic).
- Unification and proof method (conjecture: almost like for First-Order Logic).
- Constructs such as modules and embedded implications.
- Logic programming prototype.