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

François Bry

21 September 2017

Content

1 Introduction

Amalgamation (Bowen and Kowalski):

$$\mathit{believes}(\mathit{ann}, \mathit{itRains})$$
$$\mathit{believes}(\mathit{ann}, \mathit{itIsWet} \leftarrow \mathit{itRains})$$
$$\mathit{believes}(\mathit{bill}, X \leftarrow \mathit{believes}(\mathit{ann}, X))$$

In Prolog:

$$\mathit{believes}(\mathit{ann}, \mathit{itRains})$$
$$\mathit{believes}(\mathit{ann}, \mathit{cl}(\mathit{itIsWet}, \mathit{itRains}))$$
$$\mathit{believes}(\mathit{bill}, \mathit{cl}(X, \mathit{believes}(\mathit{ann}, X)))$$

1 Introduction (cont'd)

More is possible and meaningful:

$$(loves \wedge trusts)(ann, bill)$$
$$(loves \wedge trusts)(X, Y) \leftarrow loves(X, Y) \wedge trust(X, Y)$$
$$(P1 \wedge P2)(X, Y) \leftarrow P1(X, Y) \wedge P2(X, Y)$$

$$(\forall\ T\ trust(T) \Rightarrow T)(ann, bill)$$
$$(T \leftarrow trust(T))(ann, bill)$$
$$(T \leftarrow trust(T))(X, Y) \leftarrow not\ (trust(T) \wedge not\ T(X, Y))$$

1 Introduction (cont'd)

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.

2 Related Work

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.

3 Meta-Programming and Higher-Order Logics

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 propertyit defines.

Some impredicative definitons are now considered acceptable:

- Inductive definitions.
- Impredicative definitions that characterise elements of clearly apprehensible sets (including inductively defined sets).

5 Ambivalent Logic's Syntax Revisited

Ambivalent Logic (Kalsbeek and Jiang) has "amalgamation" or "impredicativity" built-in.

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

- the logical symbols consisting of the connectives $\wedge$, $\vee$, $\Rightarrow$, and $\neg$, and of the quantifiers $\forall$ and $\exists$,
- at least one (and at most finitely many) non-logical symbols each of which is distinct from every logical symbol.

5 Ambivalent Logic's Syntax Revisited (cont'd)

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: likes(ann, bill) is easily turned into ∃ X likes(ann, X) or ∀ X likes(X, bill).

6 The Barber and Russell's Paradoxes in Ambivalent Logic

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:

man(barber)

$(\forall\ y\ (man(y) \Rightarrow (shaves(barber, y) \Leftrightarrow (\neg\ shaves(y, y)))))$

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

Russell's Paradox:

$(\forall\ x\ (e(x) \Leftrightarrow (\neg\ x(x))))$

Yields the self-contradicting expression $(e(e) \Leftrightarrow (\neg\ e(e)))$

7 Ambivalent Logic's Model Theory Revisited

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

believes(bill, $\forall$ *X believes(ann, X))*
believes(bill, believes(likes(ann, bill))
student(ann)

but unlike

$\forall$ *X believes(ann, X)*
$\exists$ *X believes(ann, X)*
(student(ann) $\wedge$ *student(bill))*
(student(ann) $\Rightarrow$ *human(ann))*
student(X)

7 Ambivalent Logic's Model Theory Revisited

$\sim$: Variant relation extended to quantified expressions:

$\forall$ *X believes(ann, X)* $\sim$ $\forall$ *Y believes(ann, Y)*
believes(bill, $\forall$ *X believes(ann, X))*
$\sim$ *believes(bill,* $\forall$ *Y believes(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

believes(bill, $\forall$ *X believes(ann, X))*
believes(bill, $\forall$ *Y believes(ann, Y))*

or the class {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 $\sim$.

7 Ambivalent Logic's Model Theory Revisited (cont'd)

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.

$I(S) \models A$	iff $A \in \text{class}(A) \in S$
$I(S) \models \neg E$	iff $I(S) \not\models E$
$I(S) \models (E_1 \wedge E_2)$	iff $I(S) \models E_1$ and $I(S) \models E_2$
$I(S) \models (E_1 \vee E_2)$	iff $I(S) \models E_1$ or $I(S) \models E_2$
$I(S) \models (E_1 \Rightarrow E_2)$	iff if $I(S) \models E_1$, then $I(S) \models E_2$
$I(S) \models \exists v\ E$	iff $I(S) \models E[A/v]$ for some A
$I(S) \models \forall v\ E$	iff $I(S) \models E[A/v]$ for all A

7 Ambivalent Logic's Model Theory Revisited

In an interpretation, the quantified expressions within an atom like

believes(bill, $\forall$ *X believes(ann, X))*

are not interpreted.

Atoms like

believes(bill, $\forall$ *X believes(ann, X))*
believes(bill, $\forall$ *Y believes(ann, Y))*

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.