Substitutionless First-Order Logic: A Formal Soundness Proof [*]

Andreas Halkjær From, John Bruntse Larsen, Anders Schlichtkrull and Jørgen Villadsen

DTU Compute, AlgoLoG, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Abstract

Substitution under quantifiers is non-trivial and may obscure a proof system for newcomers. Monk (Arch. Math. Log. Grundl. 1965) successfully eliminates substitution via identities and also uses a so-called normalization of formulas as a further simplification. We formalize the substitutionless proof system in Isabelle/HOL, spelling out its side conditions explicitly and verifying its soundness.

Introduction We describe our concise formalization of the substitutionless proof system for first-order logic (Monk, 1965) in the Isabelle/HOL proof assistant (Nipkow et al., 2002). Don Monk called it the simplest known formulation.[†] Our formalization is available here: `https://bitbucket.org/isafol/isafol/src/master/FOL_Monk/`

Related Work Schlichtkrull (2018) formalizes soundness and completeness of resolution for first-order logic and gives a thorough overview of formalizations of logic found in the literature. Blanchette et al. (2017) formalize abstract soundness and completeness theorems and instantiate them for a first-order sequent calculus. Kumar et al. (2016) formalize the soundness of higher-order logic in itself (with certain assumptions). Metamath[‡] has a simple form of substitution but does not go as far as us and eliminate it entirely (Megill, 1995).

Proof System Like Monk (1965) we consider normalized formulas. Consider a predicate P of arity n. Instead of $P(x_1, x_2, \ldots, x_n)$, we write:

$$\forall v_1.\ v_1 = x_1 \to \forall v_2.\ v_2 = x_2 \to \ldots \to \forall v_n.\ v_n = x_n \to P(v_1, \ldots, v_n)$$

The order of P's arguments can be changed by changing the order of the identities and without changing the order of arguments in $P(v_1, \ldots, v_n)$. Thus the arguments to a predicate are fixed and we can represent each predicate solely by its name and arity. Monk (1965) shows that this formulation is equivalent in expressive power to ordinary first-order logic.

Axioms and Rules The proof system consists of ten axioms and two rules of inference. Variables and formulas are arbitrary unless otherwise stated.

C1	$\vdash (p \to q) \to (q \to r) \to (p \to r)$	C2	$\vdash (\neg p \to p) \to p$
C3	$\vdash p \to \neg p \to q$	C4	$\vdash (\forall x.p \to q) \to (\forall x.p) \to (\forall x.q)$
C5$_1$	$\vdash p \to (\forall x.p)$, x does not occur in p	C5$_2$	$\vdash \neg(\forall x.p) \to (\forall x.\neg(\forall x.p))$
C5$_3$	$\vdash (\forall x.\forall y.p) \to (\forall y.\forall x.p)$	C6	$\vdash \neg(\forall x.\neg\, x = y)$
C7	$\vdash x = y \to x = z \to y = z$	C8	$\vdash x = y \to p \to (\forall x.x = y \to p)$, $x \neq y$
MP	If $\vdash (p \to q)$ and $\vdash p$ then $\vdash q$	Gen	If $\vdash p$ then $\vdash \forall x.p$

[*]John Bruntse Larsen discusses a very incomplete proof in the informal publication "Certified Soundness of Simplest Known Formulation of First-Order Logic" at the ESSLLI Student Session, Toulouse, France, July 2017.

[†]And still thinks so (personal communication from Monk to Villadsen on 13 December 2016).

[‡]`http://us.metamath.org/` (see `http://us.metamath.org/#book` for the Metamath book).

The axioms make use of identities between variables instead of substitution, making for a simple proof system and formalization, even considering the normalization of formulas. The occurrence check is also very simple and only has to consider predicates and identities, disregarding quantifiers. This is made precise in the formalization below.

Formalization The proof system is formalized in Isabelle; the syntax by datatype *form*:

> **datatype** *form* = *Pre proxy arity* | *Eq var var* | *Neg form* | *Imp form form* | *Uni var form*

For predicate names we use *proxy* which is a synonym for lists of units where we only rely on the type being infinite. For clarity in the following types for functions we have introduced *arity* (predicate arities) and *var* (variable names) as synonyms for *proxy*.

The semantics is standard except for the normalized predicates. Their arguments are evaluated before the predicate is mapped to a truth value with the predicate denotation. The auxiliary function *eval* proceeds by structural recursion on the predicate's arity. The $'a$ in the types of the functions is a type variable used as the universe of quantification:

```
primrec eval :: (var ⇒ 'a) ⇒ arity ⇒ 'a list
 where
  eval - [] = [] |
  eval e (- # n) = e n # eval e n
```

```
primrec semantics :: (var ⇒ 'a) ⇒ (proxy ⇒ 'a list ⇒ bool) ⇒ form ⇒ bool
 where
  semantics e g (Pre i n) = g i (eval e n) |
  semantics e g (Eq x y) = (e x = e y) |
  semantics e g (Neg p) = (¬ semantics e g p) |
  semantics e g (Imp p q) = (semantics e g p ⟶ semantics e g q) |
  semantics e g (Uni x p) = (∀ t. semantics (e(x := t)) g p)
```

For the proof system, we here only consider the definition of axiom $\mathrm{C5}_1$. First its side condition:

```
primrec no-occur-in :: var ⇒ form ⇒ bool
 where
  no-occur-in z (Pre - n) = (length z ≥ length n) |
  no-occur-in z (Eq x y) = (z ≠ x ∧ z ≠ y) |
  no-occur-in z (Neg p) = no-occur-in z p |
  no-occur-in z (Imp p q) = (no-occur-in z p ∧ no-occur-in z q) |
  no-occur-in z (Uni - p) = no-occur-in z p
```

We note the simplicity of this definition. Next the axiom itself:

```
definition c5-1 :: var ⇒ form ⇒ form
 where
  c5-1 x p ≡ if no-occur-in x p then Imp p (Uni x p) else fail
```

We use *fail*, an abbreviation for a valid formula, when the side condition does not hold. The axioms and rules are included in an inductive definition of the derivable formulas. Isabelle allows us to also write $\vdash p$ for a derivable formula p. Soundness is then formalized as the following theorem which is proved by induction over the derivation rules:

> **theorem** *soundness*: ⊢ *p* ⟹ *semantics e g p*

The proof makes use of a few key lemmas relating the occurrence check with the semantics. One of them, called the frame property, is included below. It states that if x does not occur in p then updating its value in the environment e does not alter the semantics of p.

> **lemma** *frame-property*: *no-occur-in x p* ⟹ *semantics e g p* = *semantics* (*e*(*x* := *t*)) *g p*

Conclusion Our contribution is a formal soundness proof for a substitutionless first-order logic. Possible further work includes formalizing completeness of the proof system with respect to the semantics — a much larger undertaking — and formalizing the equivalence in expressive power between ordinary first-order logic and the normalized substitutionless version.

References

Nipkow, T., Paulson, L. C., Wenzel, M. (2002). *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Springer.

Monk, D. (1965). Substitutionless Predicate Logic with Identity. *Archiv für mathematische Logik und Grundlagenforschung*, 7 102–121.

Kumar, R. et al. (2016). Self-Formalisation of Higher-Order Logic — Semantics, Soundness, and a Verified Implementation. Journal of Automated Reasoning, 56(3) 221–259.

Blanchette, J. C., Popescu, A., Traytel, D. (2017). Soundness and Completeness Proofs by Coinductive Methods. Journal of Automated Reasoning, 58(1) 149–179.

Schlichtkrull, A. (2018). Formalization of the Resolution Calculus for First-Order Logic. Journal of Automated Reasoning, (article not yet assigned to an issue) 1–30.

Megill, N. D. (1995). A Finitely Axiomatized Formalization of Predicate Calculus with Equality. Notre Dame Journal of Formal Logic, 36(3) 435–453.