

Soundness and Completeness of an Axiomatic System for First-Order Logic

Asta Halkjær From

December 14, 2021

Abstract

This work is a formalization of the soundness and completeness of an axiomatic system for first-order logic. The proof system is based on System Q1 by Smullyan and the completeness proof follows his textbook “First-Order Logic” (Springer-Verlag 1968) [2]. The completeness proof is in the Henkin style [1] where a consistent set is extended to a maximal consistent set using Lindenbaum’s construction and Henkin witnesses are added during the construction to ensure saturation as well. The resulting set is a Hintikka set which, by the model existence theorem, is satisfiable in the Herbrand universe.

Contents

1	Syntax	3
2	Semantics	3
3	Operations	3
3.1	Shift	3
3.2	Parameters	4
3.3	Instantiation	4
3.4	Size	5
4	Propositional Semantics	5
5	Calculus	5
6	Soundness	6
7	Derived Rules	6
8	Consistent	7
9	Extension	8

10 Maximal	9
11 Saturation	9
12 Hintikka	10
12.1 Model Existence	10
12.2 Maximal Consistent Sets are Hintikka Sets	10
13 Countable Formulas	11
14 Completeness	11
15 Main Result	11

theory *FOL-Axiomatic imports HOL-Library.Countable begin*

1 Syntax

datatype (*params-tm: 'f*) *tm*
 = *Var nat* ($\langle \# \rangle$)
 | *Fun 'f* $\langle 'f \text{ tm list} \rangle$ ($\langle \dagger \rangle$)

abbreviation *Const* ($\langle \star \rangle$) **where** $\langle \star a \equiv \dagger a \ [] \rangle$

datatype (*params-fm: 'f, 'p*) *fm*
 = *Falsity* ($\langle \perp \rangle$)
 | *Pre 'p* $\langle 'f \text{ tm list} \rangle$ ($\langle \ddagger \rangle$)
 | *Imp* $\langle ('f, 'p) \text{ fm} \rangle$ $\langle ('f, 'p) \text{ fm} \rangle$ (**infixr** $\langle \longrightarrow \rangle$ 25)
 | *Uni* $\langle ('f, 'p) \text{ fm} \rangle$ ($\langle \forall \rangle$)

abbreviation *Neg* ($\langle \neg \rightarrow \rangle$ [40] 40) **where** $\langle \neg p \equiv p \longrightarrow \perp \rangle$

term $\langle \forall (\perp \longrightarrow \ddagger''P'' [\dagger''f'' [\#0]]) \rangle$

2 Semantics

definition *shift* :: $\langle (\text{nat} \Rightarrow 'a) \Rightarrow \text{nat} \Rightarrow 'a \Rightarrow \text{nat} \Rightarrow 'a \rangle$
 ($\langle \cdot \langle \cdot \rangle \rangle$ [90, 0, 0] 91) **where**
 $\langle E \langle n : x \rangle = (\lambda m. \text{if } m < n \text{ then } E \ m \text{ else if } m = n \text{ then } x \text{ else } E \ (m-1)) \rangle$

primrec *semantics-tm* ($\langle \llbracket \cdot, \cdot \rrbracket \rangle$) **where**
 $\langle \llbracket E, F \rrbracket (\#n) = E \ n \rangle$
 | $\langle \llbracket E, F \rrbracket (\dagger f \ ts) = F \ f \ (\text{map } \llbracket E, F \rrbracket \ ts) \rangle$

primrec *semantics-fm* ($\langle \llbracket \cdot, \cdot, \cdot \rrbracket \rangle$) **where**
 $\langle \llbracket \cdot, \cdot, \cdot \rrbracket \perp = \text{False} \rangle$
 | $\langle \llbracket E, F, G \rrbracket (\ddagger P \ ts) = G \ P \ (\text{map } \llbracket E, F \rrbracket \ ts) \rangle$
 | $\langle \llbracket E, F, G \rrbracket (p \longrightarrow q) = (\llbracket E, F, G \rrbracket p \longrightarrow \llbracket E, F, G \rrbracket q) \rangle$
 | $\langle \llbracket E, F, G \rrbracket (\forall p) = (\forall x. \llbracket E \langle 0 : x \rangle, F, G \rrbracket p) \rangle$

proposition $\langle \llbracket E, F, G \rrbracket (\forall (\ddagger P [\# 0]) \longrightarrow \ddagger P [\star a]) \rangle$
 $\langle \text{proof} \rangle$

3 Operations

3.1 Shift

lemma *shift-eq* [*simp*]: $\langle n = m \implies (E \langle n : x \rangle) \ m = x \rangle$
 $\langle \text{proof} \rangle$

lemma *shift-gt* [simp]: $\langle m < n \implies (E\langle n:x \rangle) m = E m \rangle$
<proof>

lemma *shift-lt* [simp]: $\langle n < m \implies (E\langle n:x \rangle) m = E (m-1) \rangle$
<proof>

lemma *shift-commute* [simp]: $\langle E\langle n:y \rangle \langle 0:x \rangle = E \langle 0:x \rangle \langle n+1:y \rangle \rangle$
<proof>

3.2 Parameters

abbreviation $\langle \text{params } S \equiv \bigcup p \in S. \text{params-fm } p \rangle$

lemma *upd-params-tm* [simp]: $\langle f \notin \text{params-tm } t \implies (\llbracket E, F(f := x) \rrbracket t = (\llbracket E, F \rrbracket t) \rangle$
<proof>

lemma *upd-params-fm* [simp]: $\langle f \notin \text{params-fm } p \implies \llbracket E, F(f := x), G \rrbracket p = \llbracket E, F, G \rrbracket p \rangle$
<proof>

lemma *finite-params-tm* [simp]: $\langle \text{finite } (\text{params-tm } t) \rangle$
<proof>

lemma *finite-params-fm* [simp]: $\langle \text{finite } (\text{params-fm } p) \rangle$
<proof>

3.3 Instantiation

primrec *lift-tm* ($\langle \uparrow \rangle$) **where**

$\langle \uparrow(\#n) = \#(n+1) \rangle$
 $\mid \langle \uparrow(\dagger f ts) = \dagger f (\text{map } \uparrow ts) \rangle$

primrec *inst-tm* ($\langle \lrcorner \llbracket - \rceil \lrcorner \rangle$) [90, 0, 0] 91) **where**

$\langle (\#n) \llbracket s/m \rrbracket = (\text{if } n < m \text{ then } \#n \text{ else if } n = m \text{ then } s \text{ else } \#(n-1)) \rangle$
 $\mid \langle (\dagger f ts) \llbracket s/m \rrbracket = \dagger f (\text{map } (\lambda t. t \llbracket s/m \rrbracket) ts) \rangle$

primrec *inst-fm* ($\langle \lrcorner \llbracket - \rceil \lrcorner \rangle$) [90, 0, 0] 91) **where**

$\langle \perp \llbracket - \rceil = \perp \rangle$
 $\mid \langle (\dagger P ts) \llbracket s/m \rrbracket = \dagger P (\text{map } (\lambda t. t \llbracket s/m \rrbracket) ts) \rangle$
 $\mid \langle (p \longrightarrow q) \llbracket s/m \rrbracket = (p \llbracket s/m \rrbracket \longrightarrow q \llbracket s/m \rrbracket) \rangle$
 $\mid \langle (\forall p) \llbracket s/m \rrbracket = \forall (p \llbracket \uparrow s/m + 1 \rrbracket) \rangle$

lemma *lift-lemma* [simp]: $\langle (\llbracket E \langle 0:x \rangle, F \rrbracket (\uparrow t) = (\llbracket E, F \rrbracket t) \rangle$
<proof>

lemma *inst-tm-semantics* [simp]: $\langle (\llbracket E, F \rrbracket (t \llbracket s/m \rrbracket)) = (\llbracket E \langle m:(\llbracket E, F \rrbracket s) \rangle, F \rrbracket t) \rangle$
<proof>

lemma *inst-fm-semantic* [simp]: $\langle \llbracket E, F, G \rrbracket (p\langle t/m \rangle) = \llbracket E\langle m:\langle E, F \rangle t \rangle, F, G \rrbracket p \rangle$
 $\langle \text{proof} \rangle$

3.4 Size

The built-in *size* is not invariant under substitution.

primrec *size-fm* **where**

$\langle \text{size-fm } \perp = 1 \rangle$
 $| \langle \text{size-fm } (\dagger -) = 1 \rangle$
 $| \langle \text{size-fm } (p \longrightarrow q) = 1 + \text{size-fm } p + \text{size-fm } q \rangle$
 $| \langle \text{size-fm } (\forall p) = 1 + \text{size-fm } p \rangle$

lemma *size-inst-fm* [simp]:
 $\langle \text{size-fm } (p\langle t/m \rangle) = \text{size-fm } p \rangle$
 $\langle \text{proof} \rangle$

4 Propositional Semantics

primrec *boolean* **where**

$\langle \text{boolean } - \perp = \text{False} \rangle$
 $| \langle \text{boolean } G - (\dagger P \text{ ts}) = G P \text{ ts} \rangle$
 $| \langle \text{boolean } G A (p \longrightarrow q) = (\text{boolean } G A p \longrightarrow \text{boolean } G A q) \rangle$
 $| \langle \text{boolean } - A (\forall p) = A (\forall p) \rangle$

abbreviation $\langle \text{tautology } p \equiv \forall G A. \text{boolean } G A p \rangle$

proposition $\langle \text{tautology } (\forall (\dagger P [\#0]) \longrightarrow \forall (\dagger P [\#0])) \rangle$
 $\langle \text{proof} \rangle$

lemma *boolean-semantic*: $\langle \text{boolean } (\lambda a. G a \circ \text{map } (\langle E, F \rangle)) \llbracket E, F, G \rrbracket = \llbracket E, F, G \rrbracket \rangle$
 $\langle \text{proof} \rangle$

lemma *tautology*: $\langle \text{tautology } p \implies \llbracket E, F, G \rrbracket p \rangle$
 $\langle \text{proof} \rangle$

proposition $\langle \exists p. (\forall E F G. \llbracket E, F, G \rrbracket p) \wedge \neg \text{tautology } p \rangle$
 $\langle \text{proof} \rangle$

5 Calculus

Adapted from System Q1 by Smullyan in First-Order Logic (1968)

inductive *Axiomatic* ($\langle \vdash \rightarrow [20] 20 \rangle$) **where**

$TA: \langle \text{tautology } p \implies \vdash p \rangle$
 $| IA: \langle \vdash \forall p \longrightarrow p\langle t/0 \rangle \rangle$
 $| MP: \langle \vdash p \longrightarrow q \implies \vdash p \implies \vdash q \rangle$

| *GR*: $\langle \vdash q \longrightarrow p \langle \star a / 0 \rangle \Longrightarrow a \notin \text{params } \{p, q\} \Longrightarrow \vdash q \longrightarrow \forall p \rangle$

lemmas

TA[*simp*]

MP[*trans, dest*]

GR[*intro*]

We simulate assumptions on the lhs of \vdash with a chain of implications on the rhs.

primrec *imply* (**infixr** $\langle \rightsquigarrow \rangle$ 26) **where**

$\langle \langle [] \rightsquigarrow q \rangle = q \rangle$

| $\langle (p \# ps \rightsquigarrow q) = (p \longrightarrow ps \rightsquigarrow q) \rangle$

abbreviation *Axiomatic-assms* ($\langle \vdash \rightarrow \rangle$ [20, 20] 20) **where**

$\langle ps \vdash q \equiv \vdash ps \rightsquigarrow q \rangle$

6 Soundness

theorem *soundness*: $\langle \vdash p \Longrightarrow \llbracket E, F, G \rrbracket p \rangle$

$\langle \text{proof} \rangle$

corollary $\langle \neg (\vdash \perp) \rangle$

$\langle \text{proof} \rangle$

7 Derived Rules

lemma *AS*: $\langle \vdash (p \longrightarrow q \longrightarrow r) \longrightarrow (p \longrightarrow q) \longrightarrow p \longrightarrow r \rangle$

$\langle \text{proof} \rangle$

lemma *AK*: $\langle \vdash q \longrightarrow p \longrightarrow q \rangle$

$\langle \text{proof} \rangle$

lemma *Neg*: $\langle \vdash \neg \neg p \longrightarrow p \rangle$

$\langle \text{proof} \rangle$

lemma *contraposition*:

$\langle \vdash (p \longrightarrow q) \longrightarrow \neg q \longrightarrow \neg p \rangle$

$\langle \vdash (\neg q \longrightarrow \neg p) \longrightarrow p \longrightarrow q \rangle$

$\langle \text{proof} \rangle$

lemma *GR'*: $\langle \vdash \neg p \langle \star a / 0 \rangle \longrightarrow q \Longrightarrow a \notin \text{params } \{p, q\} \Longrightarrow \vdash \neg \forall p \longrightarrow q \rangle$

$\langle \text{proof} \rangle$

lemma *Imp3*: $\langle \vdash (p \longrightarrow q \longrightarrow r) \longrightarrow ((s \longrightarrow p) \longrightarrow (s \longrightarrow q) \longrightarrow s \longrightarrow r) \rangle$

$\langle \text{proof} \rangle$

lemma *imply-ImpE*: $\langle \vdash ps \rightsquigarrow p \longrightarrow ps \rightsquigarrow (p \longrightarrow q) \longrightarrow ps \rightsquigarrow q \rangle$

$\langle \text{proof} \rangle$

lemma *MP'* [*trans, dest*]: $\langle ps \vdash p \longrightarrow q \Longrightarrow ps \vdash p \Longrightarrow ps \vdash q \rangle$
<proof>

lemma *imply-Cons* [*intro*]: $\langle ps \vdash q \Longrightarrow p \# ps \vdash q \rangle$
<proof>

lemma *imply-head* [*intro*]: $\langle p \# ps \vdash p \rangle$
<proof>

lemma *imply-lift-Imp* [*simp*]:
assumes $\langle \vdash p \longrightarrow q \rangle$
shows $\langle \vdash p \longrightarrow ps \rightsquigarrow q \rangle$
<proof>

lemma *add-imply* [*simp*]: $\langle \vdash q \Longrightarrow ps \vdash q \rangle$
<proof>

lemma *imply-mem* [*simp*]: $\langle p \in set\ ps \Longrightarrow ps \vdash p \rangle$
<proof>

lemma *deduct1*: $\langle ps \vdash p \longrightarrow q \Longrightarrow p \# ps \vdash q \rangle$
<proof>

lemma *imply-append* [*iff*]: $\langle (ps @ qs \rightsquigarrow r) = (ps \rightsquigarrow qs \rightsquigarrow r) \rangle$
<proof>

lemma *imply-swap-append*: $\langle ps @ qs \vdash r \Longrightarrow qs @ ps \vdash r \rangle$
<proof>

lemma *deduct2*: $\langle p \# ps \vdash q \Longrightarrow ps \vdash p \longrightarrow q \rangle$
<proof>

lemmas *deduct* [*iff*] = *deduct1 deduct2*

lemma *cut* [*trans, dest*]: $\langle p \# ps \vdash r \Longrightarrow q \# ps \vdash p \Longrightarrow q \# ps \vdash r \rangle$
<proof>

lemma *Boole*: $\langle (\neg p) \# ps \vdash \perp \Longrightarrow ps \vdash p \rangle$
<proof>

lemma *imply-weaken*: $\langle ps \vdash q \Longrightarrow set\ ps \subseteq set\ ps' \Longrightarrow ps' \vdash q \rangle$
<proof>

8 Consistent

definition $\langle consistent\ S \equiv \nexists S'. set\ S' \subseteq S \wedge (S' \vdash \perp) \rangle$

lemma *UN-finite-bound*:

assumes $\langle \text{finite } A \rangle$ **and** $\langle A \subseteq (\bigcup n. f\ n) \rangle$
shows $\langle \exists m :: \text{nat. } A \subseteq (\bigcup n \leq m. f\ n) \rangle$
 $\langle \text{proof} \rangle$

lemma *split-list*:

assumes $\langle x \in \text{set } A \rangle$
shows $\langle \text{set } (x \# \text{removeAll } x\ A) = \text{set } A \wedge x \notin \text{set } (\text{removeAll } x\ A) \rangle$
 $\langle \text{proof} \rangle$

lemma *imply-params-fm*: $\langle \text{params-fm } (ps \rightsquigarrow q) = \text{params-fm } q \cup (\bigcup p \in \text{set } ps. \text{params-fm } p) \rangle$
 $\langle \text{proof} \rangle$

lemma *inconsistent-fm*:

assumes $\langle \text{consistent } S \rangle$ **and** $\langle \neg \text{consistent } (\{p\} \cup S) \rangle$
obtains S' **where** $\langle \text{set } S' \subseteq S \rangle$ **and** $\langle p \# S' \vdash \perp \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-add-witness*:

assumes $\langle \text{consistent } S \rangle$ **and** $\langle (\neg \forall p) \in S \rangle$ **and** $\langle a \notin \text{params } S \rangle$
shows $\langle \text{consistent } (\{\neg p \langle \star a / 0 \rangle\} \cup S) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-add-instance*:

assumes $\langle \text{consistent } S \rangle$ **and** $\langle \forall p \in S \rangle$
shows $\langle \text{consistent } (\{p \langle t / 0 \rangle\} \cup S) \rangle$
 $\langle \text{proof} \rangle$

9 Extension

fun *witness where*

$\langle \text{witness used } (\neg \forall p) = \{\neg p \langle \star (\text{SOME } a. a \notin \text{used}) / 0 \rangle\} \rangle$
 $| \langle \text{witness } - - = \{\} \rangle$

primrec *extend where*

$\langle \text{extend } S\ f\ 0 = S \rangle$
 $| \langle \text{extend } S\ f\ (\text{Suc } n) =$
 $(\text{let } S_n = \text{extend } S\ f\ n \text{ in}$
 $\text{if } \text{consistent } (\{f\ n\} \cup S_n)$
 $\text{then } \text{witness } (\text{params } (\{f\ n\} \cup S_n))\ (f\ n) \cup \{f\ n\} \cup S_n$
 $\text{else } S_n) \rangle$

definition $\langle \text{Extend } S\ f \equiv \bigcup n. \text{extend } S\ f\ n \rangle$

lemma *Extend-subset*: $\langle S \subseteq \text{Extend } S\ f \rangle$
 $\langle \text{proof} \rangle$

lemma *extend-bound*: $\langle (\bigcup n \leq m. \text{extend } S\ f\ n) = \text{extend } S\ f\ m \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-params-witness* [*simp*]: $\langle \text{finite } (\text{params } (\text{witness used } p)) \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-params-extend* [*simp*]: $\langle \text{finite } (\text{params } S) \implies \text{finite } (\text{params } (\text{extend } S f n)) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-witness*:
fixes $p :: \langle ('f, 'p) \text{ fm} \rangle$
assumes $\langle \text{consistent } S \rangle$ **and** $\langle p \in S \rangle$ **and** $\langle \text{params } S \subseteq \text{used} \rangle$
and $\langle \text{finite used} \rangle$ **and** $\langle \text{infinite } (\text{UNIV} :: 'f \text{ set}) \rangle$
shows $\langle \text{consistent } (\text{witness used } p \cup S) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-extend*:
fixes $f :: \langle \text{nat} \Rightarrow ('f, 'p) \text{ fm} \rangle$
assumes $\langle \text{consistent } S \rangle$ **and** $\langle \text{finite } (\text{params } S) \rangle$
and $\langle \text{infinite } (\text{UNIV} :: 'f \text{ set}) \rangle$
shows $\langle \text{consistent } (\text{extend } S f n) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-Extend*:
fixes $f :: \langle \text{nat} \Rightarrow ('f, 'p) \text{ fm} \rangle$
assumes $\langle \text{consistent } S \rangle$ **and** $\langle \text{finite } (\text{params } S) \rangle$
and $\langle \text{infinite } (\text{UNIV} :: 'f \text{ set}) \rangle$
shows $\langle \text{consistent } (\text{Extend } S f) \rangle$
 $\langle \text{proof} \rangle$

10 Maximal

definition $\langle \text{maximal } S \equiv \forall p. p \notin S \longrightarrow \neg \text{consistent } (\{p\} \cup S) \rangle$

lemma *maximal-exactly-one*:
assumes $\langle \text{consistent } S \rangle$ **and** $\langle \text{maximal } S \rangle$
shows $\langle p \in S \longleftrightarrow (\neg p) \notin S \rangle$
 $\langle \text{proof} \rangle$

lemma *maximal-Extend*:
assumes $\langle \text{surj } f \rangle$
shows $\langle \text{maximal } (\text{Extend } S f) \rangle$
 $\langle \text{proof} \rangle$

11 Saturation

definition $\langle \text{saturated } S \equiv \forall p. (\neg \forall p) \in S \longrightarrow (\exists a. (\neg p \langle \star a / 0 \rangle) \in S) \rangle$

lemma *saturated-Extend*:

assumes $\langle \text{consistent } (\text{Extend } S f) \rangle$ **and** $\langle \text{surj } f \rangle$
shows $\langle \text{saturated } (\text{Extend } S f) \rangle$
 $\langle \text{proof} \rangle$

12 Hintikka

locale *Hintikka* =
fixes $H :: \langle ('f, 'p) \text{ fm set} \rangle$
assumes
NoFalsity: $\langle \perp \notin H \rangle$ **and**
ImpP: $\langle (p \longrightarrow q) \in H \implies p \notin H \vee q \in H \rangle$ **and**
ImpN: $\langle (p \longrightarrow q) \notin H \implies p \in H \wedge q \notin H \rangle$ **and**
UniP: $\langle \forall p \in H \implies \forall t. p \langle t/0 \rangle \in H \rangle$ **and**
UniN: $\langle \forall p \notin H \implies \exists a. p \langle \star a/0 \rangle \notin H \rangle$

12.1 Model Existence

abbreviation *hmodel* $\langle \llbracket - \rrbracket \rangle$ **where** $\llbracket H \rrbracket \equiv \llbracket \#, \dagger, \lambda P \text{ ts. Pre } P \text{ ts} \in H \rrbracket$

lemma *semantics-tm-id* [*simp*]:
 $\langle \llbracket \#, \dagger \rrbracket t = t \rangle$
 $\langle \text{proof} \rangle$

lemma *semantics-tm-id-map* [*simp*]: $\langle \text{map } \llbracket \#, \dagger \rrbracket \text{ ts} = \text{ts} \rangle$
 $\langle \text{proof} \rangle$

theorem *Hintikka-model*:
assumes $\langle \text{Hintikka } H \rangle$
shows $\langle p \in H \longleftrightarrow \llbracket H \rrbracket p \rangle$
 $\langle \text{proof} \rangle$

12.2 Maximal Consistent Sets are Hintikka Sets

lemma *inconsistent-head*:
assumes $\langle \text{consistent } S \rangle$ **and** $\langle \text{maximal } S \rangle$ **and** $\langle p \notin S \rangle$
obtains S' **where** $\langle \text{set } S' \subseteq S \rangle$ **and** $\langle p \notin S' \vdash \perp \rangle$
 $\langle \text{proof} \rangle$

lemma *inconsistent-parts* [*simp*]:
assumes $\langle ps \vdash \perp \rangle$ **and** $\langle \text{set } ps \subseteq S \rangle$
shows $\langle \neg \text{consistent } S \rangle$
 $\langle \text{proof} \rangle$

lemma *Hintikka-Extend*:
fixes $H :: \langle ('f, 'p) \text{ fm set} \rangle$
assumes $\langle \text{consistent } H \rangle$ **and** $\langle \text{maximal } H \rangle$ **and** $\langle \text{saturated } H \rangle$
and $\langle \text{infinite } (\text{UNIV} :: 'f \text{ set}) \rangle$
shows $\langle \text{Hintikka } H \rangle$
 $\langle \text{proof} \rangle$

13 Countable Formulas

instance $tm :: (countable) countable$
 $\langle proof \rangle$

instance $fm :: (countable, countable) countable$
 $\langle proof \rangle$

14 Completeness

lemma *imply-completeness*:

fixes $p :: \langle ('f :: countable, 'p :: countable) fm \rangle$
assumes $\langle \forall (E :: - \Rightarrow 'f tm) F G. Ball X \llbracket E, F, G \rrbracket \longrightarrow \llbracket E, F, G \rrbracket p \rangle$
and $\langle finite (params X) \rangle$ **and** $\langle infinite (UNIV :: 'f set) \rangle$
shows $\langle \exists ps. set ps \subseteq X \wedge (ps \vdash p) \rangle$
 $\langle proof \rangle$

theorem *completeness*:

fixes $p :: \langle (nat, nat) fm \rangle$
assumes $\langle \forall (E :: nat \Rightarrow nat tm) F G. \llbracket E, F, G \rrbracket p \rangle$
shows $\langle \vdash p \rangle$
 $\langle proof \rangle$

15 Main Result

abbreviation $valid :: \langle (nat, nat) fm \Rightarrow bool \rangle$ **where**
 $\langle valid p \equiv \forall (E :: nat \Rightarrow nat tm) F G. \llbracket E, F, G \rrbracket p \rangle$

theorem *main*: $\langle valid p \longleftrightarrow (\vdash p) \rangle$
 $\langle proof \rangle$

end

References

- [1] L. Henkin. The discovery of my completeness proofs. *Bulletin of Symbolic Logic*, 2(2):127–158, 1996.
- [2] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.