The Logiweb sequent calculus

Klaus Grue*

GRD-2006-06-21.UTC:10:43:40.552317

Contents

Abstract

The 'Logiweb sequent calculus' is suited for terse, human readable formulation of axioms, inference rules, theories, lemmas, and proofs. It supports different styles of theories like equational theories and FOL based theories. It permits to express arbitrary side conditions in the 'Logiweb programming language'. As an example of use, the calculus allows to express the deduction rule as an inference rule which makes it easy to get started using a theory. The calculus has operations for 'dereferencing' and 'referencing' which allow to convert e.g. the name of a lemma into the contents of the lemma and vice versa. The calculus interacts smoothly with fully automatic tactics as well as proofs in which part of the work is done by a human author and part is done by proof tactics.

1 Introduction

Logiweb [Gru04, Gru05, Gru06a] is a system for electronic publication of logic. It allows authors different places in the world to define theories, state and prove lemmas, and to publish *Logiweb pages* that contain the results. The present paper is an example of a Logiweb page, and the present paper is correct in the sense that it has been verified by Logiweb.

*Department of Computer Science, University of Copenhagen (DIKU)

results of each other in that a proof written by one author may make references across the internet to lemmas proved by another author. Cross theory cooperation (e.g. use of ZFC results in NBG) could be more cumbersome and cross proof system cooperation even more so, depending on the theories and systems involved.
The present paper introduces *Logiweb sequent calculus* which allows users to express arbitrary axiomatic theories on the same footing. That calculus supports arbitrary styles of logic (e.g. FOL or equational) equally well. The calculus is machine friendly in that proofs are easy to generate, verify, store, and transmit and it is general so that users of Logiweb are not tempted to make a proof system each. User friendliness in the sense that theories, lemmas, and proofs should be easy to read and write is taken care of by Logiwebs rendering and Turing complete macro expansion facilities combined with the support for proof tactics in the implementation of the calculus in [Gru06a]. The present paper gives an overview. Details are in a web appendix [Gru06b].

2 Logiweb sequent calculus

Let $\underline{v}$ and $\underline{t}$ denote the syntax classes of metavariables and object terms, respectively. The format $\underline{s}$ of Logiweb statements is $\underline{s} ::= \underline{v} \mid \underline{t} \mid \underline{s} \vdash \underline{s} \mid \underline{t} \Vdash \underline{s} \mid \forall \underline{v}\colon \underline{s} \mid \perp\!\!\!\perp \mid \underline{s} \oplus \underline{s}$. As an example of use, consider the following:

Let	A	denote	$\forall \underline{a}\colon \underline{a} + 0 = \underline{a}$.
Let	R	denote	$\forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{a} = \underline{c} \vdash \underline{b} = \underline{c}$.
Let	C	denote	$0 = 1 \vdash \perp\!\!\!\perp$.
Let	T	denote	$\mathrm{R} \oplus \mathrm{C} \oplus \mathrm{A}$.
Let	L	denote	$\mathrm{T} \vdash \forall \underline{a}\colon \underline{a} = \underline{a}$.

The construct $\forall \underline{x}\colon \underline{a}$ states that $\underline{a}$ is provable for all object terms $\underline{x}$. Hence, A above is an axiom scheme which says that $\underline{a} + 0 = \underline{a}$ for all object terms $\underline{a}$. The construct $\underline{a} \vdash \underline{b}$ states that if $\underline{a}$ is provable then $\underline{b}$ is provable. $\underline{a} \vdash \underline{b}$ is right associative and has higher priority than $\forall \underline{x}\colon \underline{a}$, so R above means $\forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{a} = \underline{c} \vdash \underline{b} = \underline{c}$. Hence, R is the inference rule which states that $\underline{a} = \underline{b}$ and $\underline{a} = \underline{c}$ infer $\underline{b} = \underline{c}$. The macro that expands $\forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{d}$ into $\forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{d}$ is defined in [Gru06b].
The construct $\perp\!\!\!\perp$ denotes absurdity, i.e. meta-falsehood (we reserve F and $\perp$ to denote falsehood and infinite looping, respectively, at the object level). Hence, C above states that $0 = 1$ is an absurdity.
The construct $\underline{a} \oplus \underline{b}$ states that both $\underline{a}$ and $\underline{b}$ are provable. Hence, T above is the axiomatic theory which comprises the axiom scheme A, the inference rule R, and the statement that $0 = 1$ is an absurdity.
The lemma L above says that $\underline{a} = \underline{a}$ is provable in the theory T.

say that $\underline{a}$ *endorses* $\underline{b}$. The endorsement operator allows to express side conditions as we shall see later.
A *Logiweb sequent* is a triple $\mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T}$ where c is a Logiweb statement and p and s are finite sets of Logiweb statements. The sequent $\{\mathsf{p}_1, \ldots, \mathsf{p}_m\} :: \{\mathsf{s}_1, \ldots, \mathsf{s}_n\} :: \mathsf{c} :: \mathsf{T}$ represents $\mathsf{p}_1 \vdash \ldots \vdash \mathsf{p}_m \vdash \mathsf{s}_1 \Vdash \ldots \Vdash \mathsf{s}_n \Vdash \mathsf{c}$. The operations of Logiweb sequent calculus are:

$$
\begin{array}{rcl}
\mathsf{a}^{\mathrm{I}} & \rightarrow & \emptyset, \emptyset, \mathsf{a} \vdash \mathsf{a} :: \mathsf{T} \\
\mathsf{a} \vdash \mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T} & \rightarrow & \mathsf{p} \backslash \{\mathsf{a}\}, \mathsf{s}, \mathsf{a} \vdash \mathsf{c} :: \mathsf{T} \\
\mathsf{a} \Vdash \mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T} & \rightarrow & \mathsf{p}, \mathsf{s} \backslash \{\mathsf{a}\}, \mathsf{a} \Vdash \mathsf{c} :: \mathsf{T} \\
\forall \mathsf{x}\colon \mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T} & \rightarrow & \mathsf{p} :: \mathsf{s} :: \forall \mathsf{x}\colon \mathsf{c} :: \mathsf{T}^{1} \\
\mathsf{p}, \mathsf{s}, \mathsf{a} \vdash \mathsf{c} :: \mathsf{T}^{\triangleright} & \rightarrow & \mathsf{p} \cup \{\mathsf{a}\} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T} \\
\mathsf{p}, \mathsf{s}, \mathsf{a} \Vdash \mathsf{c} :: \mathsf{T}^{\triangleright} & \rightarrow & \mathsf{p} :: \mathsf{s} \cup \{\mathsf{a}\} :: \mathsf{c} :: \mathsf{T} \\
\mathsf{p} :: \mathsf{s} :: \forall \mathsf{x}\colon \mathsf{c} :: \mathsf{T} \mathbin{@} \mathsf{a} & \rightarrow & \mathsf{p} :: \mathsf{s} :: \langle \mathsf{c} \,|\mathsf{x}{:=}\, \mathsf{a} \rangle :: \mathsf{T}^{2} \\
\mathsf{p}, \mathsf{s}, \mathsf{a} \Vdash \mathsf{b} :: \mathsf{T}^{\mathrm{V}} & \rightarrow & \mathsf{p} :: \mathsf{s} :: \mathsf{b} :: \mathsf{T}^{3} \\
\mathsf{p}, \mathsf{s}, \mathsf{a} \vdash \mathsf{b} \vdash \mathsf{c} :: \mathsf{T}^{+} & \rightarrow & \mathsf{p}, \mathsf{s}, \mathsf{a} \oplus \mathsf{b} \vdash \mathsf{c} :: \mathsf{T} \\
\mathsf{p}, \mathsf{s}, \mathsf{a} \oplus \mathsf{b} \vdash \mathsf{c} :: \mathsf{T}^{-} & \rightarrow & \mathsf{p}, \mathsf{s}, \mathsf{a} \vdash \mathsf{b} \vdash \mathsf{c} :: \mathsf{T} \\
\mathsf{p} :: \mathsf{s} :: \mathsf{n} :: \mathsf{T}^{*} & \rightarrow & \mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T}^{4} \\
\mathsf{p} :: \mathsf{s} :: \mathsf{c} :: \mathsf{T} \text{ i.e. } \mathsf{n} & \rightarrow & \mathsf{p} :: \mathsf{s} :: \mathsf{n} :: \mathsf{T}^{4} \\
\mathsf{p}_1 :: \mathsf{s}_1 :: \mathsf{c}_1 :: \mathsf{T}; \mathsf{p}_2 :: \mathsf{s}_2 :: \mathsf{c}_2 :: \mathsf{T} & \rightarrow & \mathsf{p}_1 \cup \mathsf{p}_2 \backslash \{\mathsf{c}_1\} :: \mathsf{s}_1 \cup \mathsf{s}_2 :: \mathsf{c}_2 :: \mathsf{T}
\end{array}
$$

Evaluation of a sequent operation gives a sequent or an exception. Exceptional cases are omitted above. Now let A' denote $\mathrm{T} \vdash \forall \underline{a}\colon \underline{a} + 0 = \underline{a}$. We have

$$
\begin{array}{l}
\mathrm{T} \vdash \mathrm{T}^{\mathrm{I}\triangleright *}; \mathrm{R} \oplus \mathrm{C} \vdash \mathrm{L}^{\mathrm{I}+*} \text{ i.e. } \mathrm{A}' \\
\mathrm{T} \vdash \emptyset, \emptyset, \mathrm{T} \vdash \mathrm{T} :: \mathrm{T}^{\triangleright *}; \mathrm{R} \oplus \mathrm{C} \vdash \emptyset, \emptyset, \mathrm{A} \vdash \mathrm{A} :: \mathrm{T}^{+*} \text{ i.e. } \mathrm{A}' \\
\mathrm{T} \vdash \{\mathrm{T}\} :: \emptyset :: \mathrm{T} :: \mathrm{T}^{*}; \emptyset, \emptyset, \mathrm{R} \oplus \mathrm{C} \vdash \mathrm{A} \vdash \mathrm{A} :: \mathrm{T}^{+*} \text{ i.e. } \mathrm{A}' \\
\mathrm{T} \vdash \{\mathrm{T}\}, \emptyset, \mathrm{R} \oplus \mathrm{C} \oplus \mathrm{A} :: \mathrm{T}; \emptyset, \emptyset, \mathrm{R} \oplus \mathrm{C} \oplus \mathrm{A} \vdash \mathrm{A} :: \mathrm{T}^{*} \text{ i.e. } \mathrm{A}' \\
\mathrm{T} \vdash \{\mathrm{T}\}, \emptyset, \mathrm{R} \oplus \mathrm{C} \oplus \mathrm{A} :: \mathrm{T}; \{\mathrm{R} \oplus \mathrm{C} \oplus \mathrm{A}\} :: \emptyset :: \mathrm{A} :: \mathrm{T}^{*} \text{ i.e. } \mathrm{A}' \\
\mathrm{T} \vdash \{\mathrm{T}\} :: \emptyset :: \mathrm{L} :: \mathrm{T}^{*} \text{ i.e. } \mathrm{A}' \rightarrow \mathrm{T} \vdash \{\mathrm{T}\} :: \emptyset :: \forall \underline{a}\colon \underline{a} + 0 = \underline{a} :: \mathrm{T} \text{ i.e. } \mathrm{A}' \\
\emptyset, \emptyset, \mathrm{T} \vdash \forall \underline{a}\colon \underline{a} + 0 = \underline{a} :: \mathrm{T} \text{ i.e. } \mathrm{A}' \rightarrow \emptyset :: \emptyset :: \mathrm{A}' :: \mathrm{T}
\end{array}
$$

which proves A'.
The syntax of *Logiweb sequent terms* reads
$\underline{q} ::= \underline{s}^{\mathrm{I}} \mid \underline{s} \vdash \underline{q} \mid \underline{s} \Vdash \underline{q} \mid \forall \underline{v}\colon \underline{q} \mid \underline{q}^{\triangleright} \mid \underline{q} \mathbin{@} \underline{s} \mid \underline{q}^{\mathrm{V}} \mid \underline{q}^{+} \mid \underline{q}^{-} \mid \underline{q} \text{ i.e. } \underline{s} \mid \underline{q}^{*} \mid \underline{q}; \underline{q}$. A sequent term q is said to *prove* the statement c if q evaluates to $\emptyset :: \overline{\emptyset} :: \mathsf{c} :: \mathsf{T}$ ([Gru06a] allows q to evaluate to $\mathsf{p} :: \emptyset :: \mathsf{c} :: \mathsf{T}$ provided all elements of p are names of previously proved lemmas).

[1] if x is not free in any member of p or s
[2] if a is free for x in c
[3] if computation of a yields T in the Logiweb programming language
[4] if n is defined to denote c

friendly, but it needs syntactic sugar. The user friendliness of the combined system of Logiweb, Logiweb sequent calculus, and syntactic sugar has been tested during a course in logic in which ten students wrote Logiweb pages that proved $\bar{x} + \bar{y} = \bar{y} + \bar{x}$ from the raw axioms of Mendelsons system S (Peano arithmetic) [Men87]. It was found that the students could prove commutativity, get the proof verified, and write a report on that in less than three weeks, starting with little experience in Logiweb, mechanical proof checking, and logic in general. For links to the reports, consult http://yoa.dk/.

While proving $\bar{x} + \bar{y} = \bar{y} + \bar{x}$ from the raw axioms may be a good student exercise, it was found that the deduction theorem and parallel instantiation of object variables was badly needed. Versions of theorems that used inference $\vdash$ and meta variables were easier to use than theorems using implication $\Rightarrow$ and object variables, but the axiom of induction forced implication and object variables upon the students. Instantiation of object variables is cumbersome because axiom A4 in [Men87] only allows to instantiate one variable at a time whereas the students needed parallel instantiation, e.g. when swapping two object variables. And hypothetical reasoning either required manual unfolding of the deduction theorem or programming of a proof tactic that could cost a lot of CPU-time for large developments.

An *inference rule of deduction* would eliminate these problems. If, however, $\forall_{\mathrm{obj}}\bar{x}\colon \forall_{\mathrm{obj}}\bar{y}\colon \bar{x} = \bar{y} \Rightarrow \bar{y} = \bar{x}$ is an object statement and $\forall \underline{a}\colon \forall \underline{b}\colon \underline{a} = \underline{b} \vdash \underline{b} = \underline{a}$ is a meta statement, then the deduction rule is a meta meta statement. We do not want to introduce a meta meta level above the Logiweb sequent calculus just to express deduction so we shall express deduction using the side condition machinery of the sequent calculus.

3.1 Peano arithmetic

A modified version of Mendelsons system S (Peano arithmetic) [Men87] may be formulated thus:

$[\mathrm{S} \overset{\mathrm{stmt}}{\rightarrow} \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} + \underline{b}' = \underline{a} + \underline{b}' \oplus \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} + \underline{b} = \underline{b} + \underline{a} \Rightarrow \underline{a} + \underline{b}' = \underline{b}' + \underline{a}][\mathrm{MP} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} + \underline{b} = \underline{b} + \underline{a} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} + \underline{b}' = \underline{b}' + \underline{a}][\mathrm{MP} \overset{\mathrm{proof}}{\rightarrow} \forall \underline{a}\colon$
Rule tactic]

$[\mathrm{Gen} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{x}\colon \forall \underline{a}\colon \underline{a} \vdash \forall_{\mathrm{obj}} \underline{x}\colon \underline{a}][\mathrm{Gen} \overset{\mathrm{proof}}{\rightarrow} \mathrm{Rule\ tactic}][\mathrm{Ded} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \lambda \mathrm{x}.\mathrm{Ded}_0(\lceil \underline{a} \rceil, \lceil \underline{b} \rceil) \Vdash \underline{a} \vdash \underline{b}][\mathrm{Ded} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic]

$[\mathrm{S2} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} = \underline{b} \vdash \underline{a}' = \underline{b}'][\mathrm{S2} \overset{\mathrm{proof}}{\rightarrow}$
Rule tactic]

$[\mathrm{S3} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \neg 0 = \underline{a}'][\mathrm{S3} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$][\mathrm{S4} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a}' = \underline{b}' \vdash \underline{a} = \underline{b}][\mathrm{S4} \overset{\mathrm{proof}}{\rightarrow}$
Rule tactic]

$[\mathrm{S7} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \underline{a} \cdot 0 = 0][\mathrm{S7} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$ $[\mathrm{S8} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} \cdot \underline{b}' = \underline{a} \cdot \underline{b} + \underline{a}][\mathrm{S8} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$

$[\mathrm{Neg} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \neg \underline{b} \Rightarrow \neg \underline{a} \vdash \neg \underline{b} \Rightarrow \underline{a} \vdash \underline{b}][\mathrm{Neg} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$

$[\mathrm{S1} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{a} = \underline{c} \vdash \underline{b} = \underline{c}][\mathrm{S1} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$

$[\mathrm{S9} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{x}\colon \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \langle \lceil \underline{b} \rceil \equiv^0 \lceil \underline{a} \rceil \| \lceil \underline{x} \rceil := \lceil 0 \rceil \rangle \Vdash \langle \lceil \underline{c} \rceil \equiv^0 \lceil \underline{a} \rceil \| \lceil \underline{x} \rceil := \lceil \underline{x}' \rceil \rangle \Vdash \underline{b} \vdash \underline{a} \Rightarrow \underline{c} \vdash \underline{a}][\mathrm{S9} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$[5]

As defined in [Gru06a], $[\mathrm{S5} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \underline{a} + 0 = \underline{a}][\mathrm{S5} \overset{\mathrm{proof}}{\rightarrow}$ Rule tactic$]$ macro expands into a lemma and a proof where the lemma says $\mathrm{S} \vdash \forall \underline{a}\colon \underline{a} + 0 = \underline{a}$ and the proof is a proof of that lemma.
$[\mathrm{S} \overset{\mathrm{stmt}}{\rightarrow} \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} + \underline{b}' = \underline{a} + \underline{b}' \oplus \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} \Rightarrow \underline{b} \vdash \underline{a} \vdash \underline{b} \oplus \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} = \underline{b} \vdash \underline{a}' = \underline{b}' \oplus \forall \underline{a}\colon \forall \underline{b}\colon \underline{a}' = \underline{b}' \vdash \underline{a} = \underline{b} \oplus \forall \underline{a}\colon \forall \underline{b}\colon \lambda x.\mathrm{Ded}_0(\lceil \underline{a} \rceil, \lceil \underline{b} \rceil) \Vdash \underline{a} \vdash \underline{b} \oplus \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} \cdot \underline{b}' = \underline{a} \cdot \underline{b} + \underline{a} \oplus \forall \underline{a}\colon \underline{a} + 0 = \underline{a} \oplus \forall \underline{a}\colon \forall \underline{b}\colon \neg \underline{b} \Rightarrow \neg \underline{a} \vdash \neg \underline{b} \Rightarrow \underline{a} \vdash \underline{b} \oplus \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{a} = \underline{c} \vdash \underline{b} = \underline{c} \oplus \forall \underline{x}\colon \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \langle \lceil \underline{b} \rceil \equiv^0 \lceil \underline{a} \rceil \| \lceil \underline{x} \rceil := \lceil 0 \rceil \rangle \Vdash \langle \lceil \underline{c} \rceil \equiv^0 \lceil \underline{a} \rceil \| \lceil \underline{x} \rceil := \lceil \underline{x}' \rceil \rangle \Vdash \underline{b} \vdash \underline{a} \Rightarrow \underline{c} \vdash \underline{a} \oplus \forall \underline{a}\colon \neg 0 = \underline{a}' \oplus \forall \underline{x}\colon \forall \underline{a}\colon \underline{a} \vdash \forall_{\mathrm{obj}} \underline{x}\colon \underline{a} \oplus \forall \underline{a}\colon \underline{a} \cdot 0 = 0]$ macro expands into a definition which defines S as the conjunction of $\forall \underline{a}\colon \underline{a} + 0 = \underline{a}$ and all the other rules attributed to S. The $[\mathcal{X} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{x}]$ macro is somewhat complex since it has to scan the entire page to find all rules related to the theory being defined. The benefit of collecting rules from the entire page is that it gives authors the freedom to introduce axioms one by one.
The deduction rule Ded is such that e.g. $\forall \underline{a}\colon \forall \underline{b}\colon \underline{a} = \underline{b} \vdash \underline{b} = \underline{a}$ allows to conclude $\underline{a} = \underline{b} \Rightarrow \underline{b} = \underline{a}$ and $\overline{\mathrm{x}} + \overline{\mathrm{y}} = \overline{\mathrm{y}} + \overline{\mathrm{x}}$ allows to conclude $\underline{a} + \underline{b} = \underline{b} + \underline{a}$ so Ded implements both deduction and parallel instantiation. All complexity is hidden in the side condition which is defined in [Gru06b].
Having the deduction rule, axioms A1, A2, A4, and A5 in [Men87] become superfluous. For proofs of those axioms based on deduction, see [Gru06b].
As an example of a development, [Gru06b] proves the following lemmas from [Men87] in system S:

$[\mathrm{Prop\ 3.2a} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \underline{a} = \underline{a}]$

$[\mathrm{Prop\ 3.2b} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \underline{a} = \underline{b} \vdash \underline{b} = \underline{a}]$

$[\mathrm{Prop\ 3.2c} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{b} = \underline{c} \vdash \underline{a} = \underline{c}]$

$[\mathrm{Prop\ 3.2d} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{c} \vdash \underline{b} = \underline{c} \vdash \underline{a} = \underline{b}]$

$[\mathrm{Prop\ 3.2e} \overset{\mathrm{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{a}\colon \forall \underline{b}\colon \forall \underline{c}\colon \underline{a} = \underline{b} \vdash \underline{a} + \underline{c} = \underline{b} + \underline{c}]$

[5] $\langle \lceil \underline{a} \rceil \equiv^0 \lceil \underline{b} \rceil \| \lceil \underline{x} \rceil := \lceil \underline{c} \rceil \rangle$ says 'the object term $\underline{b}$ where the object variable $\underline{x}$ is replaced by the object term $\underline{c}$ is alpha equivalent to the object term $\underline{a}$, c.f. [Gru06b]

$[\text{Prop } 3.2\text{h} \overset{\text{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{\mathsf{a}}\colon \forall \underline{\mathsf{b}}\colon \underline{\mathsf{a}} + \underline{\mathsf{b}} = \underline{\mathsf{b}} + \underline{\mathsf{a}}]$

$[\text{Prop } 3.2\text{i} \overset{\text{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{\mathsf{a}}\colon \forall \underline{\mathsf{b}}\colon \forall \underline{\mathsf{c}}\colon \underline{\mathsf{a}} = \underline{\mathsf{b}} \vdash \underline{\mathsf{c}} + \underline{\mathsf{a}} = \underline{\mathsf{c}} + \underline{\mathsf{b}}]$

In the present paper we merely show the proof of an auxiliary lemma which proves the induction step of Prop 3.2f:

$[\text{Prop } 3.2\text{f}_2 \overset{\text{stmt}}{\rightarrow} \mathrm{S} \vdash \forall \underline{\mathsf{a}}\colon \underline{\mathsf{a}} = 0 + \underline{\mathsf{a}} \Rightarrow \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}']$

$[\text{Prop } 3.2\text{f}_2 \overset{\text{proof}}{\rightarrow} \lambda \mathsf{c}.\lambda \mathsf{x}.\mathcal{P}(\lceil \mathrm{S} \vdash \forall \underline{\mathsf{a}}\colon \forall \underline{\mathsf{a}}\colon \underline{\mathsf{a}} = 0 + \underline{\mathsf{a}} \vdash \mathrm{S2} \rhd \underline{\mathsf{a}} = 0 + \underline{\mathsf{a}} \gg \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}'; \mathrm{S6} \gg 0 + \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}'; \text{Prop } 3.2\text{d} \rhd \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}' \rhd 0 + \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}' \gg \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}'; \mathrm{Ded} \rhd \forall \underline{\mathsf{a}}\colon \underline{\mathsf{a}} = 0 + \underline{\mathsf{a}} \vdash \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}' \gg \underline{\mathsf{a}} = 0 + \underline{\mathsf{a}} \Rightarrow \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}' \rceil, \mathrm{p}_0, \mathsf{c})]$

As defined in [Gru06a], $\forall \underline{\mathsf{a}}\colon \underline{\mathsf{b}}$ macro expands into $\forall \underline{\mathsf{a}}\colon \underline{\mathsf{b}}$.
$\mathrm{S2} \rhd \mathrm{L04} \gg \underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}'; \underline{\mathsf{b}}$ macro expands into a local macro definition and a call to a proof tactic. The local macro definition defines L05 as shorthand for $\underline{\mathsf{a}}' = 0 + \underline{\mathsf{a}}'$ and the proof tactic expands the line into $\mathrm{S2}^{\mathrm{I} \rhd * \rhd} \mathbin{@} \underline{\mathsf{a}} \mathbin{@} 0 + \underline{\mathsf{a}}; \underline{\mathsf{b}}$ using unification. For more details see [Gru06b].

4 Conclusion and further work

Logiweb sequent calculus with examples from traditional Peano arithmetic has been presented. For further examples, click 'Map Theory' at http://yoa.dk/. Map theory is an equational theory. In that theory, all definitions present on a lemmas home page become axioms according to an 'axiom of definition' which is expressible by a side condition in the Logiweb sequent calculus.
The traditional deduction theorem requires that no application of Gen is made to variables free in the premise. The side condition of the deduction rule above is quite different. Proving the consistency of the deduction rule is work of the future but is expected to be straightforward since, in the standard model, a statement holds for all natural numbers if and only if it holds for all terms.

References

[Gru04] K. Grue. Logiweb. In Fairouz Kamareddine, editor, *Mathematical Knowledge Management Symposium 2003*, volume 93 of *Electronic Notes in Theoretical Computer Science*, pages 70–101. Elsevier, 2004.

[Gru05] K. Grue. The implementation of logiweb. In Bernd Fischer, Stephan Schulz, and Geoff Sutcliffe, editors, *Empirically Successful Classical Automated Reasoning (ESCAR)*, 2005.

[Gru06b] K. Grue. Logiweb sequent calculus, appendix. Technical report, Logiweb, 2006. http://www.diku.dk/cgi-bin/cginetd/grue/relay/go/0126D93B3C690A5DBF88BD72AE26B798B3A982EE2DBD8DEAFA8C82A30806/2/body/tex/appendix.pdf.

[Men87] E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth and Brooks, 3. edition, 1987.