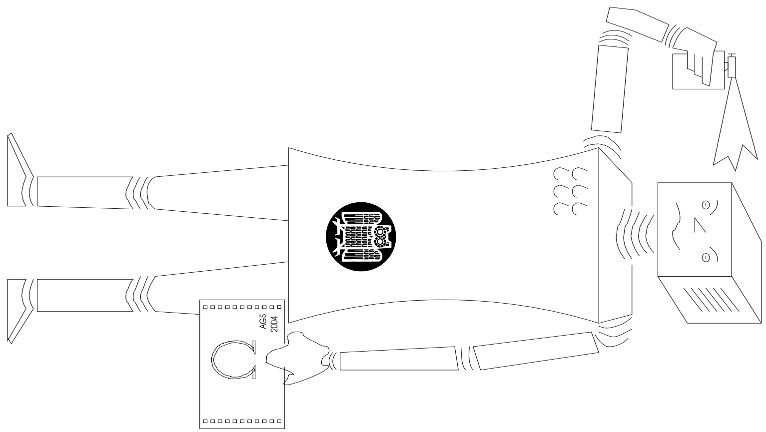
SEKI-Working-Paper ISSN 1860-5931

UNIVERSITÄT DES SAARLANDES
FACHRICHTUNG INFORMATIK
D–66123 SAARBRÜCKEN
GERMANY
WWW: http://www.ags.uni-sb.de/

Generalization is Semantical

Claus-Peter Wirth
FR Informatik, Saarland Univ.
chris@ags.uni-sb.de

SEKI-Working-Paper SWP–2005–02



This SEKI-Working-Paper was internally reviewed by:
Erica Melis
FR Informatik, Universität des Saarlandes, D–66123 Saarbrücken, Germany
E-mail: melis@activemath.org
WWW: http://www.activemath.org/~melis/

Editor of SEKI series:
Claus-Peter Wirth
FR Informatik, Universität des Saarlandes, D–66123 Saarbrücken, Germany
E-mail: cp@ags.uni-sb.de
WWW: http://www.ags.uni-sb.de/~cp/welcome.html

Generalization is Semantical

Claus-Peter Wirth
FR Informatik, Saarland Univ.
chris@ags.uni-sb.de

April 14, 2005

Abstract

We want to list some examples that show that it is hopeless to achieve generalization of inductive theorems by syntactical means.

1 Specification Too Poor

The following example is from Kühler (2000), p.11.

Suppose we have the definitions

(double1) $\mathsf{double}(0){=}0$
(double2) $\mathsf{double}(\mathsf{s}(x^{\delta^-})){=}\mathsf{s}(\mathsf{s}(\mathsf{double}(x^{\delta^-})))$

and want to prove the clause

(double-irrefl) $\mathsf{double}(x^{\delta^-}){\neq}x^{\delta^-},\ \ x^{\delta^-}{=}0$

This cannot be proved without the introduction of additional functions such as

(less1) $\mathsf{less}(0,\mathsf{s}(y^{\delta^-})){=}\mathsf{true}$
(less2) $\mathsf{less}(x^{\delta^-},0){=}\mathsf{false}$
(less3) $\mathsf{less}(\mathsf{s}(x^{\delta^-}),\mathsf{s}(y^{\delta^-})){=}\mathsf{less}(x^{\delta^-},y^{\delta^-})$

to formulate lemmas such as

(less-s) $\mathsf{less}(x^{\delta^-},\mathsf{s}(x^{\delta^-})){=}\mathsf{true}$
(less-trans) $\mathsf{less}(x^{\delta^-},z^{\delta^-}){=}\mathsf{true},\ \ \mathsf{less}(x^{\delta^-},y^{\delta^-}){\neq}\mathsf{true},\ \ \mathsf{less}(y^{\delta^-},z^{\delta^-}){\neq}\mathsf{true}$
(less-double) $\mathsf{less}(x^{\delta^-},\mathsf{double}(x^{\delta^-})){=}\mathsf{true},\ \ x^{\delta^-}{=}0$
(less-$\neq$) $\mathsf{less}(x^{\delta^-},y^{\delta^-}){=}\mathsf{false},\ \ x^{\delta^-}{\neq}y^{\delta^-}$

Now (less-s) and (less-trans) are trivial by induction and prove (less-double) by induction, and this together with the inductively trivial (less-$\neq$) deductively implies (double-irrefl).

It seems obvious that the generation of all these symbols, definitions, and lemmas cannot be reasonably hoped to be achieved by syntactic manipulations in any framework.

2 Properly Generalizing does Not Find a Proof

The example of this section is a rational reconstruction and elaboration of Section 5.2 of Busch (1993) within the superior formalism of Wirth (2004). We do not agree with the conclusion of Section 5.2 of Busch (1993):

> "Although another induction proof has to be performed, the complexity of the induction compared to the original goal is reduced."

To the contrary, we do not see a reduction in complexity because a direct induction proof of the open goals is still not possible. We see this as an example, where generalization helps to find a theorem but not a proof of it.

Suppose we have the definitions

(append1) $\mathsf{append}(\mathsf{nil}, l_2^{\delta^-}){=}l_2^{\delta^-}$
(append2) $\mathsf{append}(\mathsf{cons}(x_1^{\delta^-}, l_1^{\delta^-}), l_2^{\delta^-}){=}\mathsf{cons}(x_1^{\delta^-}, \mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}))$
(revtail1) $\mathsf{revtail}(\mathsf{nil}, l_2^{\delta^-}){=}l_2^{\delta^-}$
(revtail2) $\mathsf{revtail}(\mathsf{cons}(x_1^{\delta^-}, l_1^{\delta^-}), l_2^{\delta^-}){=}\mathsf{revtail}(l_1^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_2^{\delta^-}))$

and want to prove

(revtail-append1) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), \mathsf{nil}){=}\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, \mathsf{nil}), \mathsf{revtail}(l_1^{\delta^-}, \mathsf{nil}))$

Although this is inductively valid, a simple proof fails because the induction hypothesis is too weak. Both straightforward generalizations

(revtail-append2) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-}){=}\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, l_3^{\delta^-}), \mathsf{revtail}(l_1^{\delta^-}, l_3^{\delta^-}))$
(revtail-append3) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-}){=}\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, l_4^{\delta^-}), \mathsf{revtail}(l_1^{\delta^-}, l_5^{\delta^-}))$

can easily be refuted, however. To generalize avoiding over-generalization, we should try to prove

(revtail-append4) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\gamma}(l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})){=}$
$\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})), \mathsf{revtail}(l_1^{\delta^-}, l_5^{\gamma}(l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})))$

with variable-condition $\{l_3^{\gamma}, l_4^{\gamma}, l_5^{\gamma}\} \times \{l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}\}$. As induction over $l_1^{\delta^-}$ is unflawed, let us start it. The case $\{l_1^{\delta^-} \mapsto \mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-})\}$ adds $\{l_1^{\delta^-}\} \times \{x_1^{\delta^-}, l_6^{\delta^-}\}$ to our variable-condition and reduces to

(revtail-append4.2) $\mathsf{revtail}(\mathsf{append}(l_6^{\delta^-}, l_2^{\delta^-}), \mathsf{cons}(x_1^{\delta^-}, l_3^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}))){=}$

$$\mathsf{append}\begin{pmatrix}\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})),\\ \mathsf{revtail}(l_6^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_5^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})))\end{pmatrix}$$

Unifying the left hand side of this with the left-hand side of the induction hypothesis (revtail-append4) under the substitution $\{l_1^{\delta^-}{\mapsto}l_6^{\delta^-}\}$ would require

$$\mathsf{cons}(x_1^{\delta^-}, l_3^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})){=}l_3^{\gamma}(l_6^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}),$$

which is hopeless. But with a substitution that replaces all unmatched free δ^--variables with free γ-variables, say $\{l_1^{\delta^-}{\mapsto}l_6^{\delta^-},\ l_3^{\delta^-}{\mapsto}l_{3'}^{\gamma},\ l_4^{\delta^-}{\mapsto}l_{4'}^{\gamma},\ l_5^{\delta^-}{\mapsto}l_{5'}^{\gamma}\}$, this requirement becomes

$$\mathsf{cons}(x_1^{\delta^-}, l_3^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})){=}l_3^{\gamma}(l_6^{\delta^-}, l_2^{\delta^-}, l_{3'}^{\gamma}, l_{4'}^{\gamma}, l_{5'}^{\gamma}),$$

which, after substitution $\{l_3^{\gamma} \mapsto \lambda l_1, l_2, l_3, l_4, l_5.\ l_3\}$ and $\lambda\beta$-reduction, reads

$$\mathsf{cons}(x_1^{\delta^-}, l_3^{\delta^-}){=}l_{3'}^{\gamma},$$

which is solved by $\{l_{3'}^{\gamma} \mapsto \mathsf{cons}(x_1^{\delta^-}, l_3^{\delta^-})\}$. After this the original clause reads

$$\begin{aligned}\text{(revtail-append4)}\quad & \mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-})= \\ & \mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})), \mathsf{revtail}(l_1^{\delta^-}, l_5^{\gamma}(l_1^{\delta^-}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})))\end{aligned}$$

and the result of the induction hypothesis application is

$$\begin{aligned}\text{(revtail-append4.2.1)}\quad & \mathsf{append}\begin{pmatrix}\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(l_6^{\delta^-}, l_2^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_3^{\delta^-}), l_{4'}^{\gamma}, l_{5'}^{\gamma})), \\ \mathsf{revtail}(l_6^{\delta^-}, l_5^{\gamma}(l_6^{\delta^-}, l_2^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_3^{\delta^-}), l_{4'}^{\gamma}, l_{5'}^{\gamma}))\end{pmatrix}= \\ & \mathsf{append}\begin{pmatrix}\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})), \\ \mathsf{revtail}(l_6^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_5^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})))\end{pmatrix}\end{aligned}$$

As there are many choices to close this goal by instantiation and $\lambda\beta$-reduction, let us have a look at the base case first. The case $\{l_1^{\delta^-} \mapsto \mathsf{nil}\}$ reduces to

$$\text{(revtail-append4.1)}\quad \mathsf{revtail}(l_2^{\delta^-}, l_3^{\delta^-})=\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, l_4^{\gamma}(\mathsf{nil}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})), l_5^{\gamma}(\mathsf{nil}, l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}))$$

Now induction on $l_2^{\delta^-}$ is unflawed. The base case $\{l_2^{\delta^-} \mapsto \mathsf{nil}\}$ reduces to

$$\text{(revtail-append4.1.1)}\quad l_3^{\delta^-}=\mathsf{append}(l_4^{\gamma}(\mathsf{nil}, \mathsf{nil}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}), l_5^{\gamma}(\mathsf{nil}, \mathsf{nil}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}))$$

Rewriting with (append1), this (unsafely) reduces to

$$\begin{aligned}&\text{(revtail-append4.1.1.1)}\quad l_3^{\delta^-}=l_5^{\gamma}(\mathsf{nil}, \mathsf{nil}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}) \\ &\text{(revtail-append4.1.1.2)}\quad l_4^{\gamma}(\mathsf{nil}, \mathsf{nil}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})=\mathsf{nil}\end{aligned}$$

As the substitution $\{l_5^{\gamma} \mapsto \lambda l_1, l_2, l_3, l_4, l_5.\ l_3^{\delta^-}\}$ is blocked by our variable-condition, (revtail-append4.1.1.1) can only be closed by applying $\{l_5^{\gamma} \mapsto \lambda l_1, l_2, l_3, l_4, l_5.\ l_3\}$ and $\lambda\beta$-reduction. With this, (revtail-append4.2.1) can be (unsafely) reduced to

$$\text{(revtail-append4.2.1.1)}\quad l_4^{\gamma}(l_6^{\delta^-}, l_2^{\delta^-}, \mathsf{cons}(x_1^{\delta^-}, l_3^{\delta^-}), l_{4'}^{\gamma}, l_{5'}^{\gamma})=l_4^{\gamma}(\mathsf{cons}(x_1^{\delta^-}, l_6^{\delta^-}), l_2^{\delta^-}, l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})$$

The case $\{l_2^{\delta^-} \mapsto \mathsf{cons}(x_2^{\delta^-}, l_7^{\delta^-})\}$ of (revtail-append4.1) adds $\{l_2^{\delta^-}\} \times \{x_2^{\delta^-}, l_7^{\delta^-}\}$ to our variable-condition and reduces to

$$\begin{aligned}\text{(revtail-append4.1.2)}\quad & \mathsf{revtail}(l_7^{\delta^-}, \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta}))= \\ & \mathsf{append}(\mathsf{revtail}(l_7^{\delta^-}, \mathsf{cons}(x_2^{\delta^-}, l_4^{\gamma}(\mathsf{nil}, \mathsf{cons}(x_2^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}))), l_3^{\delta^-})\end{aligned}$$

The case $\{l_7^{\delta^-} \mapsto \mathsf{nil}\}$ reduces to

$$\text{(revtail-append4.1.2.1)}\quad \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta})=\mathsf{append}(l_4^{\gamma}(\mathsf{nil}, \mathsf{cons}(x_2^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-}), \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta^-}))$$

By (append1) this (unsafely) reduces to

$$\text{(revtail-append4.1.2.1.1)}\quad l_4^{\gamma}(\mathsf{nil}, \mathsf{cons}(x_2^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-}, l_4^{\delta^-}, l_5^{\delta^-})=\mathsf{nil}$$

Now (revtail-append4.1.1.2) and (revtail-append4.1.2.1.1) can be solved by projection on the first component and the imitation of nil, i.e. $\{l_4^{\gamma} \mapsto \lambda l_1, l_2, l_3, l_4, l_5.\ \mathsf{nil}\}$. As only the latter closes (revtail-append4.2.1.1), we apply it globally. The top open goal then is

$$\text{(revtail-append4.1)}\quad \mathsf{revtail}(l_2^{\delta^-}, l_3^{\delta^-})=\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, \mathsf{nil}), l_3^{\delta^-})$$

But again there is no chance to prove this by an easy induction. Indeed, even with the additional lemma

$$\text{(append-assoc)}\quad \mathsf{append}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-})=\mathsf{append}(l_1^{\delta^-}, \mathsf{append}(l_2^{\delta^-}, l_3^{\delta^-}))$$

Tobias Schmidt-Samoa was not able to prove (revtail-append4.1) with NQTHM, cf. Boyer & Moore (1988). The inductive theorem prover QUODLIBET, cf. Avenhaus &al. (2003), which is stronger than the famous and excellent NQTHM in the generation of induction hypotheses, proves it automatically by induction on $l_2^{\delta^-}$. We have to apply (revtail-append4.1) as an induction hypothesis *twice* to (revtail-append4.1.2) $\mathsf{revtail}(l_7^{\delta^-}, \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta}))=\mathsf{append}(\mathsf{revtail}(l_7^{\delta^-}, \mathsf{cons}(x_2^{\delta^-}, \mathsf{nil})), l_3^{\delta^-})$

namely once with $\{l_2^{\delta^-} \mapsto l_7^{\delta^-},\ l_3^{\delta^-} \mapsto \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta})\}$ and once with $\{l_2^{\delta^-} \mapsto l_7^{\delta^-},\ l_3^{\delta^-} \mapsto \mathsf{cons}(x_2^{\delta^-}, \mathsf{nil})\}$, both for rewriting. This results in

(revtail-append4.1.2.1) $\mathsf{append}(\mathsf{revtail}(l_7^{\delta^-}, \mathsf{nil}), \mathsf{cons}(x_2^{\delta^-}, l_3^{\delta}))$
$=\mathsf{append}(\mathsf{append}(\mathsf{revtail}(l_7^{\delta^-}, \mathsf{nil}), \mathsf{cons}(x_2^{\delta^-}, \mathsf{nil})), l_3^{\delta^-})$

the right-hand side of which can be rewritten to the left-hand side by application of (append-assoc), (append1), and (append1).

All in all, although we have been able to synthesize a sound conjecture, namely

(revtail-append4) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-})=\mathsf{append}(\mathsf{revtail}(l_2^{\delta^-}, \mathsf{nil}), \mathsf{revtail}(l_1^{\delta^-}, l_3^{\delta^-}))$

we did not know how to prove it. Although QuodLibet was finally able to prove (revtail-append4.1) with some additional lemma, neither I nor Nqthm were able to do it. In our opinion, this demonstrates that generalization by syntactic means is poor. Now what does a human expert do? First he sees

(revtail-append5) $\mathsf{revtail}(\mathsf{append}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-})=\mathsf{revtail}(l_2^{\delta^-}, \mathsf{revtail}(l_1^{\delta^-}, l_3^{\delta^-}))$
(revtail-append6) $\mathsf{revtail}(l_1^{\delta^-}, \mathsf{append}(l_2^{\delta^-}, l_3^{\delta^-}))=\mathsf{append}(\mathsf{revtail}(l_1^{\delta^-}, l_2^{\delta^-}), l_3^{\delta^-})$

which are trivial by induction on $l_1^{\delta^-}$ and (together with (append1)) prove (revtail-append4) and—a fortiori—(revtail-append1) by deductive rewriting, where (revtail-append6) has to be applied from right to left.

Note that the proof of (revtail-append5) (whose left-hand side is identical to the one of (revtail-append4)) is what let us instantiate l_3^{γ} in the above proof attempt due to the erroneous assumption that (revtail-append4) should be proved by induction on $l_1^{\delta^-}$.

References

Jürgen Avenhaus, Ulrich Kühler, Tobias Schmidt-Samoa, Claus-Peter Wirth (2003). *How to Prove Inductive Theorems?* QuodLibet*!*. 19th CADE 2003, LNAI 2741, pp. 328–333, Springer.

Robert S. Boyer, J S. Moore (1988). *A Computational Logic Handbook.* Academic Press.

Holger Busch (1993). *Unification-Based Induction.* In: Claesen & Gordon (1993), pp. 97–116.

Luc J. M. Claesen, Michaël J. C. Gordon (eds.) (1993). *Higher Order Logic Theorem Proving and its Applications.* North-Holland (Elsevier).

Ulrich Kühler (2000). *A Tactic-Based Inductive Theorem Prover for Data Types with Partial Operations.* Ph.D. thesis, Infix, Sankt Augustin.

Claus-Peter Wirth (2004). *Descente Infinie + Deduction.* Logic J. of the IGPL **12**, pp. 1–96, Oxford Univ. Press. `http://www.ags.uni-sb.de/~cp/p/d/welcome.html` (Sept. 12, 2003).