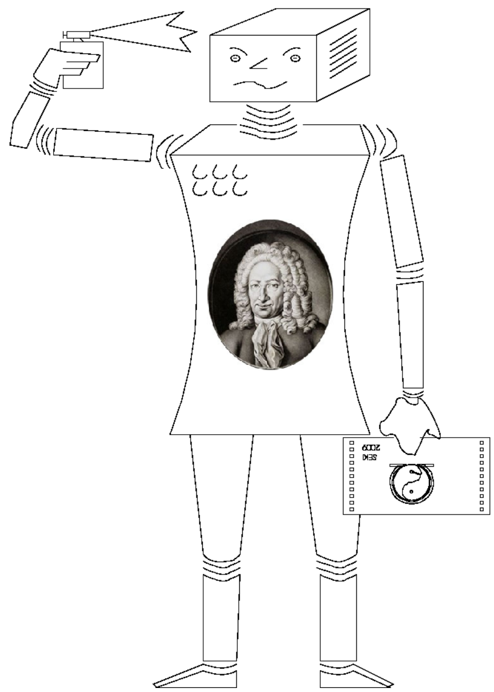
SEKI Working-Paper

http://www.ags.uni-sb.de/~cp/seki.html
ISSN 1860-5931



The P and Q Example in Resolution

Claus-Peter Wirth
Max Planck Institute for Informatics,
D–66123 Saarbrücken, Germany
wirth@logic.at

SEKI Working-Paper SWP–2009–03

SEKI is published by the following institutions:

German Research Center for Artificial Intelligence (DFKI GmbH), Germany

- Robert Hooke Str. 5, D–28359 Bremen
- Trippstadter Str. 122, D–67663 Kaiserslautern
- Campus D 3 2, D–66123 Saarbrücken

International University of Germany, Campus 1, D–76646 Bruchsal, Germany

Jacobs University Bremen, School of Engineering & Science, Campus Ring 1, D–28759 Bremen, Germany

Universität des Saarlandes, FR 6.2 Informatik, Campus, D–66123 Saarbrücken, Germany

SEKI Editor:

Claus-Peter Wirth
E-mail: `wirth@logic.at`
WWW: `http://www.ags.uni-sb.de/~cp`

Please send surface mail exclusively to:

DFKI Bremen GmbH
Safe and Secure Cognitive Systems
Cartesium
Enrique Schmidt Str. 5
D–28359 Bremen
Germany

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`

The P and Q Example in Resolution

Claus-Peter Wirth
Max Planck Institute for Informatics, D–66123 Saarbrücken, Germany
wirth@logic.at

May 11, 2009

Abstract

We solve the problem of kissing an angel effectively and more efficiently than voodoo man. Some reading of Immanuël Kant, however, is required as a precondition.

Keywords: German Idealism, Dyslexia, Kissing, Angels, Angles, Kant, MSC 03D20, MSC 68Q42, MSC 16S15

1 Introduction

We want to improve the arithmetical power of the higher-order resolution prover LEO-II. Adding the second-order axioms of the specification of the natural numbers either by Peano or by Pieri leads to an explosion of the search space which guarantees that even the most trivial theorems cannot be proved automatically. Thus it is wise to proceed in a similar fashion as in the first-order inductive theorem provers. As the full strength of the heuristics of the school of explicit induction is applicable in the *descente infinie* framework of the inductive theorem prover QUODLIBET [Avenhaus &al., 2003], but the historical restrictions of the explicit induction approach are overcome, we want to start by finding out how the fully automated part of QUODLIBET can be reimplemented in a framework of refutation resolution and paramodulation.

The P *&* Q example of [Wirth, 2004, § 3.2.2] is a challenge for the power of mutual induction. Moreover, as it is a toy example, it serves well as a first test case.

2 The P & Q Example in QuodLibet

The toy example of this § 2 illustrates how mutual induction works in the *descente infinie* framework of QuodLibet. As the proof requires mutual induction with non-trivial weights, it cannot be performed in many inductive theorem proving systems, nor in the lean induction calculus of [Baaz &al., 1997].

We use zero $\mathsf{0} : \mathsf{nat}$ and successor $\mathsf{s} : \mathsf{nat} \to \mathsf{nat}$ as constructors for the type nat. In QuodLibet we would declare the two symbols as constructors of the sort nat. In our more general context here, we explicitly have to provide the definition of the definedness predicate that singles out those elements of the sort nat that actually are natural numbers and no error elements.

(Defnat1) $\quad \mathrm{Def}(\mathsf{0})$
(Defnat2) $\quad \forall X.\ \ (\ \mathrm{Def}(\mathsf{s}(X)) \Leftarrow \mathrm{Def}(X)\ )$
(nat1) $\quad \forall X.\ \ (\ (X{=}\mathsf{0} \vee \exists Y.\ (X{=}\mathsf{s}(Y) \wedge \mathrm{Def}(Y))) \Leftarrow \mathrm{Def}(X)\ )$

This basic signature is enriched with the predicates $\mathsf{P} : \mathsf{nat} \to \mathsf{bool}$ and $\mathsf{Q} : \mathsf{nat} \to \mathsf{nat} \to \mathsf{bool}$. We have the following axioms, defining the special predicates of our example.

(P1) $\quad \mathsf{P}(\mathsf{0})$
(P2) $\quad \forall X.\ \ (\ \mathsf{P}(\mathsf{s}(X)) \Leftarrow (\ \mathsf{P}(X) \wedge \mathsf{Q}(X, \mathsf{s}(X)) \wedge \mathrm{Def}(X)\ )\ )$

(Q1) $\quad \forall X.\ \ (\ \mathsf{Q}(X, \mathsf{0}) \Leftarrow \mathrm{Def}(X)\ )$
(Q2) $\quad \forall X, Y.\ \ (\ \mathsf{Q}(X, \mathsf{s}(Y)) \Leftarrow (\ \mathsf{Q}(X, Y) \wedge \mathsf{P}(X) \wedge \mathrm{Def}(X) \wedge \mathrm{Def}(Y)\ )\ )$

We want to show that both predicates are tautological on the actual natural numbers:

(1) $\quad \mathsf{P}(X_0^{\delta^-}),\ \neg\mathrm{Def}(X_0^{\delta^-});\ \ w_1^{\gamma}(X_0^{\delta^-})$

(2) $\quad \mathsf{Q}(Y_0^{\delta^-}, Z_0^{\delta^-}),\ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \neg\mathrm{Def}(Z_0^{\delta^-});\ \ w_2^{\gamma}(Y_0^{\delta^-}, Z_0^{\delta^-})$

Note that weights consist only of weight *terms* (such as $w_1^{\gamma}(X_0^{\delta^-})$ in (1)), but not of additional induction orderings and quasi-orderings, because we fix these orderings to be the single ones of the QuodLibet system, as discussed in [Wirth, 2004, § 3.2.1]. Therefore — as discussed in [Wirth, 2004, § 2.5] — the items (3)–(6) of Theorem 2.51 of [Wirth, 2004] can be omitted in the following.

In the Hypothesizing steps for (1) and (2) we introduce the variable-condition

$$R \ := \left(\begin{array}{ll} & \mathcal{V}_{\gamma\delta^+}((1)) \times \mathcal{V}_{\delta^-}((1)) \\ \cup & \mathcal{V}_{\gamma\delta^+}((2)) \times \mathcal{V}_{\delta^-}((2)) \end{array} \right) \ = \ \left(\begin{array}{ll} & \{w_1^{\gamma}\} \times \{X_0^{\delta^-}\} \\ \cup & \{w_3^{\gamma}\} \times \{Y_0^{\delta^-}, Z_0^{\delta^-}\} \end{array} \right)$$

to have all free δ^--variables of (1) or (2) in the set Y of Theorem 2.51 of [Wirth, 2004], so that we can locally instantiate all of them with whatever we want in each application of (1) or (2) as a lemma or as an induction hypothesis.

After several inference steps, QuodLibet presents a sequent tree for (1), which is — *mutandis mutatis* — similar to following:

(1) $\mathsf{P}(X_0^{\delta^-}),\ \neg\mathrm{Def}(X_0^{\delta^-});\ w_1^{\gamma}(X_0^{\delta^-})$

$(\mathsf{nat1}), \gamma, \beta, \delta^-, \mathrm{Rewrite}^+$

(1.1) $\mathsf{P}(\mathsf{0}),\ \neg\mathrm{Def}(\mathsf{0});\ w_1^{\gamma}(\mathsf{0})$

(1.2) $\mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-}))$

$(\mathsf{P1})$

$(\mathsf{P2}), \gamma, \beta, \beta$

(1.2.1) $\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-}))$

(1.2.2) $\mathsf{Q}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})),\ \neg\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-}))$

induction-hypothesis appl. of (1)$\{X_0^{\delta^-}\mapsto X_1^{\delta^-}\}$

induction hypothesis application of (2)$\{Y_0^{\delta^-}\mapsto X_1^{\delta^-},\ Z_0^{\delta^-}\mapsto \mathsf{s}(X_1^{\delta^-})\}$

(1.2.1.1) $w_1^{\gamma}(X_1^{\delta^-}) < w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-})),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-}))$

(1.2.2.1) $w_2^{\gamma}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})) < w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-})),\ \mathsf{Q}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})),\ \neg\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta^-}))$

The square boxes are the nodes of the proof tree, whereas the round-edged boxes show applications of inference rules of Theorem 2.49 and Theorem 2.51 of [Wirth, 2004], which are more elementary than the inference rules in QuodLibet. We can check whether the tree is closed simply by checkeing whether all leaves are round-edged nodes. This is not only useful for implementation purposes (where we have to record somewhere why a branch is closed) but also immediately realizes the explicit representation of leaves required by Definition 2.42 of [Wirth, 2004].

For example, " $(\mathsf{nat1}), \gamma, \beta, \delta^-, \mathrm{Rewrite}^+$ " in the first round-edged box means that we use the axiom $(\mathsf{nat1})$ as a lemma in Theorem 2.51 of [Wirth, 2004], and then apply a γ-, a β-, and a δ^--step and several Rewrite-steps of Theorem 2.49 to get the nodes (1.1) and (1.2).

This means that, by application of (nat1),

(1) $\mathsf{P}(X_0^{\delta^-}),\ \neg\mathrm{Def}(X_0^{\delta^-});\ w_1^\gamma(X_0^{\delta^-})$

reduces to the following two sequences[1]

(1.1) $\mathsf{P}(\mathsf{0}),\ \neg\mathrm{Def}(\mathsf{0});\ w_1^\gamma(\mathsf{0})$
(1.2) $\mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$

with the additional variable-condition of $\{(X_0^{\delta^-}, X_1^{\delta^-})\}$. (1.1) is subsumed by (P1). Applying (P2) to (1.2) we get the following two clauses:

(1.2.1) $\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$
(1.2.2) $\mathsf{Q}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})),\ \neg\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$

Applying (1) to (1.2.1) as an induction hypothesis, instantiated via $\{X_0^{\delta^-}\mapsto X_1^{\delta^-}\}$, results in

(1.2.1.1) $w_1^\gamma(X_1^{\delta^-}) < w_1^\gamma(\mathsf{s}(X_1^{\delta^-})),\ \mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$

Applying (2) to (1.2.2) as an induction hypothesis, instantiated via $\{Y_0^{\delta^+}\mapsto X_1^{\delta^+},\ Z_0^{\delta^+}\mapsto \mathsf{s}(X_1^{\delta^+})\}$, results in[3]

(1.2.2.1) $w_2^\gamma(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})) < w_1^\gamma(\mathsf{s}(X_1^{\delta^-})),$
$\mathsf{Q}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})),\ \neg\mathsf{P}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$

[1] Let us have a look at this step in more detail. After applying the substitution $\{X\mapsto X_0^{\delta^-}\}$ in a γ-step; a β-step and a δ^--step (introducing $X_1^{\delta^-}$) leave us with the following two cases:
(1.1′) $X_0^{\delta^-} \neq \mathsf{0},\ \mathsf{P}(X_0^{\delta^-}),\ \neg\mathrm{Def}(X_0^{\delta^-});\ w_1^\gamma(X_0^{\delta^-})$
(1.2′) $X_0^{\delta^-} \neq \mathsf{s}(X_1^{\delta^-}),\ \mathsf{P}(X_0^{\delta^-}),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(X_0^{\delta^-});\ w_1^\gamma(X_0^{\delta^-})$
and the additional variable-condition of $\{(X_0^{\delta^-}, X_1^{\delta^-})\}$. Contextual rewriting with the head literals results in
(1.1″) $X_0^{\delta^-} \neq \mathsf{0},\ \mathsf{P}(\mathsf{0}),\ \neg\mathrm{Def}(\mathsf{0});\ w_1^\gamma(\mathsf{0})$
(1.2″) $X_0^{\delta^-} \neq \mathsf{s}(X_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(X_1^{\delta^-})),\ \neg\mathrm{Def}(X_1^{\delta^-}),\ \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))$
Now $X_0^{\delta^-}$ is in solved form[2] in both clauses, so that removing the literal containing it is an equivalence transformation.

[2] $x^{\delta^-} \in \mathrm{V}_{\delta^-}$ is in *solved form* in the weighted sequent $\Gamma\ (x^{\delta^-}\neq t)\ \Pi;\ \beth$ if $x^{\delta^-} \notin \mathcal{V}(t, \Gamma\Pi, \beth)$ and $\mathcal{V}_{\gamma\delta^+}(t, \Gamma\Pi, \beth) \subseteq R^+\langle\{x^{\delta^-}\}\rangle$.

[3] Let us have a closer look on what happens below (1.2.2). We instantiate the meta-variables of Theorem 2.51 of [Wirth, 2004] as follows:

$$\begin{array}{rcl} \Phi & := & \mathsf{Q}(Y_0^{\delta^-}, Z_0^{\delta^-}) \\ \daleth & := & w_2^\gamma(Y_0^{\delta^-}, Z_0^{\delta^-}) \\ Y & := & \{Y_0^{\delta^-}, Z_0^{\delta^-}\} \\ \varrho & := & \{Y_0^{\delta^-}\mapsto X_1^{\delta^-},\ Z_0^{\delta^-}\mapsto \mathsf{s}(X_1^{\delta^-})\} \\ M & := & \{\neg\mathsf{Q}(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})),\ \mathrm{Def}(X_1^{\delta^-}),\ \mathrm{Def}(\mathsf{s}(X_1^{\delta^-})),\ w_2^\gamma(X_1^{\delta^-}, \mathsf{s}(X_1^{\delta^-})) < w_1^\gamma(\mathsf{s}(X_1^{\delta^-}))\} \end{array}$$

For (2) we get a sequent tree very similar to that of (1):

$$(2)\quad \mathsf{Q}(Y_0^{\delta^-}, Z_0^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(Z_0^{\delta^-});\ \ w_2^{\gamma}(Y_0^{\delta^-}, Z_0^{\delta^-})$$

$$(\mathsf{nat1}), \gamma, \beta, \delta, \mathrm{Rewrite}^{+}$$

$$(2.1)\quad \mathsf{Q}(Y_0^{\delta^-}, \mathsf{0}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{0});\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{0})$$

$$(2.2)\quad \mathsf{Q}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \neg\mathrm{Def}(Z_1^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{s}(Z_1^{\delta^-}));\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-}))$$

$$(\mathsf{Q1})$$

$$(\mathsf{Q2}), \gamma, \beta, \beta, \beta$$

$$(2.2.1)\quad \mathsf{Q}(Y_0^{\delta^-}, Z_1^{\delta^-}),\ \ \mathsf{Q}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \neg\mathrm{Def}(Z_1^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{s}(Z_1^{\delta^-}));\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-}))$$

$$(2.2.2)\quad \mathsf{P}(Y_0^{\delta^-}),\ \ \neg\mathsf{Q}(Y_0^{\delta^-}, Z_1^{\delta^-}),\ \ \mathsf{Q}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \neg\mathrm{Def}(Z_1^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{s}(Z_1^{\delta^-}));\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-}))$$

ind.-hyp. appl. of $(2)\{Z_0^{\delta^-} \mapsto Z_1^{\delta^-}\}$

ind.-hyp. appl. of $(1)\{X_0^{\delta^-} \mapsto Y_0^{\delta^-}\}$

$$(2.2.1.1)\quad w_2^{\gamma}(Y_0^{\delta^-}, Z_1^{\delta^-}) < w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \mathsf{Q}(Y_0^{\delta^-}, Z_1^{\delta^-}),\ \ \mathsf{Q}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \neg\mathrm{Def}(Z_1^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{s}(Z_1^{\delta^-}));\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-}))$$

$$(2.2.2.1)\quad w_2^{\gamma}(Y_0^{\delta^-}) < w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \mathsf{P}(Y_0^{\delta^-}),\ \ \neg\mathsf{Q}(Y_0^{\delta^-}, Z_1^{\delta^-}),\ \ \mathsf{Q}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-})),\ \ \neg\mathrm{Def}(Z_1^{\delta^-}),\ \ \neg\mathrm{Def}(Y_0^{\delta^-}),\ \ \neg\mathrm{Def}(\mathsf{s}(Z_1^{\delta^-}));\ \ w_2^{\gamma}(Y_0^{\delta^-}, \mathsf{s}(Z_1^{\delta^-}))$$

We have applied each of the two weighted sequents (1) and (2) in each of their two proof trees. Luckily we used induction hypothesis application instead of lemma application. The latter would have resulted in a lemma application relation of $\{1,2\} \times \{1,2\}$ which is not well-founded and our proof trees would have been useless because we would never be able to apply Theorem 2.45 of [Wirth, 2004]. As we have used induction hypothesis application instead of lemma application, we have produced the four additional leaves (1.2.1.1), (1.2.2.1), (2.2.1.1), and (2.2.2.1), which are still open. We choose our 2nd order weight functions according to $w_1^{\gamma}(X) := (X)$ and $w_2^{\gamma}(X, Y) := (X, Y)$, or — more precisely — by applying the substitution $\{\ w_1^{\gamma} \mapsto \lambda X.\,(X),\ w_2^{\gamma} \mapsto \lambda X, Y.\,(X, Y)\ \}$ and by $\lambda\beta$-reduction, using the lexicographic combination of [Wirth, 2004, § 3.2.1], explained below.

Now the proof attempt can be successfully completed: For example, the weighted sequent (1.2.1.1) turns into

(1.2.1.1) $(X_1^{\delta-}){<}(\mathsf{s}(X_1^{\delta-}))$, $\mathsf{P}(X_1^{\delta-})$, $\mathsf{P}(\mathsf{s}(X_1^{\delta-}))$, $\neg\mathrm{Def}(X_1^{\delta-})$, $\neg\mathrm{Def}(\mathsf{s}(X_1^{\delta-}))$; $w_1^{\gamma}(\mathsf{s}(X_1^{\delta-}))$

which is equivalent to

(1.2.1.1.1) $X_1^{\delta-}{<}\mathsf{s}(X_1^{\delta-})$, $\mathsf{P}(X_1^{\delta-})$, $\mathsf{P}(\mathsf{s}(X_1^{\delta-}))$, $\neg\mathrm{Def}(X_1^{\delta-})$, $\neg\mathrm{Def}(\mathsf{s}(X_1^{\delta-}))$; $w_1^{\gamma}(\mathsf{s}(X_1^{\delta-}))$

which again is subsumed by one of QuodLibet's ordering axioms.

QuodLibet realizes a version of the *semantic length* ordering. Thus, any clause of the form

$$t{<}t', \mathit{\Gamma}$$

is a valid axiom provided that all the following items hold:

1. Each constant symbol in t or t' is a constructor.
2. Each variable X in t or t' satisfies the following:
   (a) X is of a basic type inductively defined via constructors, and
   (b) $\mathit{\Gamma}$ contains a literal of the form $\neg\mathrm{Def}(X)$.
3. No variable occurs more often in t than in t'.
4. The number of symbol occurrences in t is less than the number of symbols in t'.

Besides that QuodLibet has an additional sort for the lexicographic combination of this ordering. Note that as any completed proof will be a finite object, there is a maximum length of the ordering tuples occurring in it, so that the lexicographic combination of the well-founded semantic length ordering is well-founded indeed. Also note, that this is not generally the case: $(\mathsf{s}(0)) > (0, \mathsf{s}(0)) > (0, 0, \mathsf{s}(0)) > \ldots$. All in all, this means that term formation over this additional sort for the lexicographic combination must be restricted; for instance, the formation of the term $0^n\,\mathsf{s}(0)$ of the above example must be excluded.

The axioms of the lexicographic combination can then easily be formalized, although special heuristic treatment might be necessary for efficiency.

$$(){<}(X, L)$$
$$(X, L){<}(X', L') \quad \Leftrightarrow \quad \textbf{if } X{=}X' \textbf{ then } L{<}L' \textbf{ else } X{<}X' \textbf{ fi}$$
$$L{\not<}()$$

Which steps in this proof were typical for *inductive* theorem proving in the sense that their soundness relies on notions of inductive validity instead of the stronger notion of validity in all models?

> Besides the four induction hypothesis applications, the final branch closure rules for <-literals are typical for induction because they require that, in all models in K, the class of models to be considered for inductive validity, the successor of each natural number is different from that natural number and each natural number is built-up from zero by a finite number of successor steps (i.e. there are neither cycles nor **Z**-chains in the models, cf. [Enderton, 1972]).

3 How we cannot automate this proof in LEO-II

Why adding the second-order axioms of Peano and Pieri fails w.r.t. automation.

4 How we hope to be able to automate such proofs in LEO-II

First, just as in QUODLIBET, we have to apply a heuristics called *recursion analysis*, going back to the school of *explicit induction*, cf. [Boyer & Moore, 1979], [Walther, 1994], [Bundy, 1999]. We will take the code for this out of the code for recursion analysis in QML (QUODLIBET Meta Language).

This will tell us that it is a good idea

- to prove (1.2.1), (1.2.2) and that the proof of this is sufficient for the validity of (1); and
- to prove (2.2.1), (2.2.2) and that the proof of this is sufficient for the validity of (2).

Such a proof cannot be found without (1) and (2) available as induction hypotheses. Thus, we need something like

(1hyp) $$\mathsf{P}(X_0^{\delta-}),\ \neg\mathrm{Def}(X_0^{\delta-}),\ w_1^{\gamma}(X_0^{\delta-})\not< W_0^{\delta-};\ W_0^{\delta-}$$

and

(2hyp) $$\mathsf{Q}(Y_0^{\delta-}, Z_0^{\delta-}),\ \neg\mathrm{Def}(Y_0^{\delta-}),\ \neg\mathrm{Def}(Z_0^{\delta-}),\ w_2^{\gamma}(Y_0^{\delta-}, Z_0^{\delta-})\not< W_1^{\delta-};\ W_1^{\delta-}$$

where the "; $W_i^{\delta-}$" now does not stand for a weight term, but for a clausal constraint in LEO-II. In any resolution or paramodulation step, where one such constraint is present, this constraint must be inherited. In any step where several such constraints are present, these constraints must be unified and inherited.

This problem solved, to prove the four above goals, LEO-II has to negate them and to derive a refutation. Let us see, how (1.2.1) would look like negated: Luckily, we have $w_1^{\gamma} R^{+} X_1^{\delta-}$. This means that (1.2.1) is logically equivalent to

(1.2.1)' $$\mathsf{P}(X_1^{\delta+}) \vee \mathsf{P}(\mathsf{s}(X_1^{\delta+})) \vee \neg\mathrm{Def}(X_1^{\delta+}) \vee \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta+}));\qquad w_1^{\gamma}(\mathsf{s}(X_1^{\delta+}))$$

with choice-condition

$$C(X_1^{\delta+}) := \neg\big(\ \mathsf{P}(X_1^{\delta+}) \vee \mathsf{P}(\mathsf{s}(X_1^{\delta+})) \vee \neg\mathrm{Def}(X_1^{\delta+}) \vee \neg\mathrm{Def}(\mathsf{s}(X_1^{\delta+}))\ \big)$$

Note that the free δ^--variable $X_1^{\delta-}$ has been replace with the fresh free δ^+-variable $X_1^{\delta+}$ in a δ^+-step, which can either be seen as Skolemization or better as an application of Hilbert's ε [Wirth, 2008].

Now (1.2.1)' can be easily negated, resulting in the clauses:

$$\neg\mathsf{P}(X_1^{\delta+});\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta+}))$$
$$\neg\mathsf{P}(\mathsf{s}(X_1^{\delta+}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta+}))$$
$$\mathrm{Def}(X_1^{\delta+});\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta+}))$$
$$\mathrm{Def}(\mathsf{s}(X_1^{\delta+}));\ w_1^{\gamma}(\mathsf{s}(X_1^{\delta+}))$$

Resolving the first one with (1hyp) results in

$$\neg\mathrm{Def}(X_0^{\delta^+}),\;\; w_1^\gamma(X_0^{\delta^+})\not< w_1^\gamma(\mathsf{s}(X_1^{\delta^+}));\;\; w_1^\gamma(\mathsf{s}(X_1^{\delta^+}))$$

and resolving this with the third one results in

$$w_1^\gamma(X_0^{\delta^+})\not< w_1^\gamma(\mathsf{s}(X_1^{\delta^+}));\;\; w_1^\gamma(\mathsf{s}(X_1^{\delta^+}))$$

5 Important Aspects

Why does this work? Induction always takes place only on the outermost universal variables. In the free variable framework of [Wirth, 2004; 2008], these are those free δ^--variables X^{δ^-} of a sequent Γ that satisfy $w^\gamma R^+ X^{\delta^-}$ for each $w^\gamma \in \mathcal{V}_\gamma(\Gamma)$, where R denotes the current variable-condition. These replacing these variables with free δ^+-variables with the appropriate variable-condition and choice-condition is an equivalence transformation. Assuming that no other free δ^+-variables occur in γ, we can negate this easily. Note that our negation of the conjecture does not take place on top level, i.e. we do not state that the conjecture is (C, R)-invalid in all structures $\mathcal{A} \in \mathrm{K}$, but only that, for some $\mathcal{A} \in \mathrm{K}$ and some $(\mathcal{A}, R)$-valuation e and for some π that is $(e, \mathcal{A})$-compatible with (C, R), the conjecture is not $(\pi, e, \mathcal{A})$-valid. Thus, if we can derive the empty clause by resolution, we know that this cannot be the case w.r.t. to the information gathered in the meanwhile in the variable-condition R, the choice-condition C, via instantiation of rigid variables.

Thus, although it is a good idea for heuristic propagation to let the several refutations run in parallel, they communicate their results via the rigid variables.

Thus, if one refutation instantiates w_1^γ, this will become visible to all the other refutations carrying the proof dept for our mutual induction.

References

[Avenhaus &al., 2003] Jürgen Avenhaus, Ulrich Kühler, Tobias Schmidt-Samoa, and Claus-Peter Wirth. How to prove inductive theorems? QuodLibet! In *19th Int. Conf. on Automated Deduction, Miami Beach (FL), 2003*, number 2741 in Lecture Notes in Artificial Intelligence, pages 328–333. Springer, 2003. http://www.ags.uni-sb.de/~cp/p/quodlibet.

[Baaz &al., 1997] Matthias Baaz, Uwe Egly, and Christian G. Fermüller. Lean induction principles for tableaus. In Didier Galmiche, editor, *6th Int. Conf. on Tableaus and Related Methods, Pont-à-Mousson (France), 1997*, number 1227 in Lecture Notes in Artificial Intelligence, pages 62–75. Springer, 1997.

[Boyer & Moore, 1979] Robert S. Boyer and J S. Moore. *A Computational Logic*. Academic Press (Elsevier), 1979.

[Bundy, 1999] Alan Bundy. *The Automation of Proof by Mathematical Induction*. Informatics Research Report No. 2, Division of Informatics, Univ. Edinburgh, 1999. Also in [Robinson & Voronkow, 2001, Vol. 1, pp. 845–911].

[Enderton, 1972] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press (Elsevier), 1972.

[Gabbay &al., 1994] Dov Gabbay, Christopher John Hogger, and J. Alan Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 2: Deduction Methodologies*. Oxford Univ. Press, 1994.

[Robinson & Voronkow, 2001] Alan Robinson and Andrei Voronkow, editors. *Handbook of Automated Reasoning*. Elsevier, 2001.

[Walther, 1994] Christoph Walther. Mathematical induction. 1994. In [Gabbay &al., 1994, pp. 127–228].

[Wirth, 2004] Claus-Peter Wirth. Descente Infinie + Deduction. *Logic J. of the IGPL*, 12:1–96, 2004. http://www.ags.uni-sb.de/~cp/p/d.

[Wirth, 2008] Claus-Peter Wirth. Hilbert's epsilon as an operator of indefinite committed choice. *J. Applied Logic*, 6:287–317, 2008. http://www.ags.uni-sb.de/~cp/p/epsi, http://dx.doi.org/10.1016/j.jal.2007.07.009.