On the Mechanization of the Proof of Hessenberg's Theorem in Coherent Logic

Bezem, M.; Hendriks, R.D.A.

published in
Journal of Automated Reasoning
2008

DOI (link to publisher)
10.1007/s10817-007-9086-x

document version
Publisher's PDF, also known as Version of record

J Autom Reasoning (2008) 40:61–85
DOI 10.1007/s10817-007-9086-x

On the Mechanization of the Proof of Hessenberg's Theorem in Coherent Logic

Marc Bezem · Dimitri Hendriks

Received: 26 April 2007 / Accepted: 25 September 2007 / Published online: 29 November 2007

Abstract We propose to combine interactive proof construction with proof automation for a fragment of first-order logic called Coherent Logic (CL). CL allows enough existential quantification to make Skolemization unnecessary. Moreover, CL has a constructive proof system based on forward reasoning, which is easy to automate and where standardized proof objects can easily be obtained. We have implemented in Prolog a CL prover which generates Coq proof scripts. We test our approach with a case study: Hessenberg's theorem, which states that in elementary projective plane geometry Pappus' axiom implies Desargues' axiom. Our CL prover makes it possible to automate large parts of the proof, in particular taking care of the large number of degenerate cases.

Keywords Coherent logic · Automated theorem proving · Proof objects · Hessenberg's theorem

1 Introduction

The main purposes of automated theorem proving are *consolidation* (a formal proof increases our confidence in the theorem) and *experimentation* (testing whether some formula is a theorem). For the purpose of consolidation it is particularly relevant to have some standardized format for the proofs, preferably based on typed lambda calculus (where proof checking is type checking is decidable, see for example [10]).

M. Bezem (✉)
Department of Computer Science, University of Bergen, P.O. Box 7800,
5020 Bergen, Norway
e-mail: bezem@ii.uib.no

D. Hendriks
Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081a,
1081 HV Amsterdam, The Netherlands
e-mail: diem@cs.vu.nl

Standardized proof objects allow independent verification of proofs by other systems, as well as reuse of proofs. Further purposes of proofs are *explanation* (a proof explains why the theorem is true, and a formal proof does so in great detail) and *algorithm extraction* (depends on the logic in question). Of course the explanatory virtues of formal proofs decrease with their size, but on a more abstract level even large formal proofs can be useful in analyzing, for example, which axioms actually have been used. In all these cases success is subject to (at least) machine limitations, where the size of the search space is usually the biggest problem.

General-purpose automated theorem provers are not yet able to prove difficult mathematical theorems, in any case not in first-order logic (more expressive logics are even harder to automate). Moreover, proof objects are usually somewhat neglected and are certainly not generated in a standardized format. The formal verification of difficult theorems still requires human ingenuity, and this situation is not likely to change any time soon.

In the combined effort of man and machine called *interactive proof construction* one would like to have as much automation as possible. It turns out that boosting automation in interactive proof construction is difficult. One reason is that the best general-purpose provers first translate the problem, typically to conjunctive normal form, and then work on the translated problem. The correctness of this procedure relies on metatheorems that are quite complicated from the standpoint of proof theory. Converting a proof of the translated problem into a proof of the original problem is possible but difficult [4, 8], and the resulting proof objects are seldom satisfactory (too large and indirect by the translation). Moreover, if the translated problem isn't easily solved, steering a prover that works on a different problem is hard.

The biggest step in the translation usually is Skolemization, the elimination of existential quantifiers in favor of Skolem functions. This involves an extension of the signature and preserves validity only under some extra axioms, weak instances of the axiom of choice called 'Skolem axioms'. Extending the signature with a Skolem function can turn a finite Herbrand universe with only constants into an infinite one. Important properties of Skolem functions, such as symmetry or idempotency, can get lost in translation.

We propose to combine interactive proof construction with proof automation for a fragment of first-order logic (FOL) called *coherent logic* (CL). CL allows enough existential quantification to make Skolemization unnecessary and has a natural proof system based on forward reasoning. Moreover, this proof system can be automated in such a way that standardized proof objects are easily obtained. In case the search space is too large, forward reasoning is straightforward to steer. We test our approach by formally proving Hessenberg's theorem (1905), which states that in elementary projective plane geometry Pappus' axiom implies Desargues' axiom. Besides being a beautiful theorem, it has an interesting history in that the proof contained a gap for almost 50 years. This makes it worthwhile to formalize the proof by Cronheim [7], which was claimed to be (and indeed is) complete.

In the next section we introduce CL. In Section 3 we work toward a machine-oriented axiomatization of projective plane geometry and point out some subtleties with respect to the formulation. Section 4 exhibits an example of how the reasoning mechanism works. The complete proof of Hessenberg's theorem, assembled from three large machine-generated subproofs, is described in Section 5. We compare

our approach to related research in Section 6. We have added an Appendix about a misformulation of Desargues' axiom by Skolem, detected during our experiments, demonstrating our machinery in full detail. All files concerning the formal verification in Coq [21] using CL can be found on [3]. A preliminary version of this work was presented at ADG'06, Pontevedra, Spain.

2 Coherent Logic

As far as we know, Skolem [18] was the first who used CL (*avant la lettre*) to solve a decision problem in lattice theory and to prove the independence of Desargues' axiom from the basic axioms of projective plane geometry. Modern CL arose in algebraic geometry, see for example [12, Sect. D.1.1]. In this paper we define CL as the fragment of FOL consisting of implicitly universally quantified implications of the following form:

$$A_1 \wedge \cdots \wedge A_n \Rightarrow \exists \vec{x}_1.\, \boldsymbol{C}_1 \vee \cdots \vee \exists \vec{x}_m.\, \boldsymbol{C}_m, \tag{1}$$

where the A_i are first-order atoms and the $\boldsymbol{C}_j$ are conjunctions of such atoms. We use some obvious notational optimizations to improve readability: if $n = 0$, then we leave out $\Rightarrow$ altogether; if $m = 0$, then we write $\bot$ (*falsum*) to denote the empty disjunction; empty existential quantifications are left out. A coherent theory is a finite set of formulas of the form (1). Closed atoms, that is, atoms without free variables, are called *facts*.

Let T be a coherent theory. The set $\Delta^T(X \vdash F)$ of derivations in T of a fact F from a set of facts X is inductively defined by the following two rules (explained below).

$$\frac{X}{F} F \in X \qquad \frac{X \quad \mathbf{A} \Rightarrow \mathbf{D} \quad \delta_1 \cdots \delta_m}{F} \mathbf{A} \subseteq X$$

In words, the base case (left) applies when the goal F is in the set of facts X. The step case applies when $\boldsymbol{A} \Rightarrow \boldsymbol{D}$ is a closed instance of a formula in T whose antecedent is satisfied by X, expressed by $\boldsymbol{A} \subseteq X$. The number of subderivations δ_i is equal to the length of the disjunction $\boldsymbol{D}$. If $\boldsymbol{D} = \exists \vec{x}_1.\, \boldsymbol{C}_1 \vee \cdots \vee \exists \vec{x}_m.\, \boldsymbol{C}_m$, every subderivation δ_i should be in $\Delta^T(X, \overline{\boldsymbol{C}_i} \vdash F)$ $(1 \leq i \leq m)$. Here $X, \overline{\boldsymbol{C}_i}$ is the set of facts X extended with the atoms in $\boldsymbol{C}_i$, with the variables $\vec{x}_i$ replaced by fresh constants. If $\boldsymbol{D}$ is $\bot$, there are no subderivations.

As an example we give the derivation of r in the coherent theory with axioms $p \vee \exists x.q(x)$, $p \Rightarrow \bot$, $q(x) \Rightarrow r$.

$$\frac{\varnothing \quad p \vee \exists x.q(x) \qquad \dfrac{\{p\} \quad p \Rightarrow \bot}{r} \qquad \dfrac{\{q(c)\} \quad q(c) \Rightarrow r \qquad \dfrac{\{q(c), r\}}{r}}{r}}{r}$$

The rightmost inference is the only base case; all other inferences are step cases. Note that the step using $p \Rightarrow \bot$ has no subderivations but is not a base case.

There are two views on CL's proof theory. One is to view the above rules as a natural deduction-style system of inference, which leads via the Curry–Howard

correspondence to a standard format for proofs. The step case is actually a combination of instantiation (of the formulas in T), Modus ponens (combining X with $\boldsymbol{A} \Rightarrow \boldsymbol{D}$), disjunction elimination (the subderivations for each disjunct in $\boldsymbol{D}$), and existential elimination (the fresh constants in $\overline{\boldsymbol{C}_i}$).

The other view is that of forward reasoning. This can also be observed in the example above: we start with the empty set of facts; we split in two branches with sets $\{p\}$ and $\{q(c)\}$, respectively; and finally extend the latter to $\{q(c), r\}$. Observe that the axioms are used as production rules (Skolem: *Erzeugungsprinzipien*) to generate new facts from already known ones, distinguishing cases for each disjunct in the consequent and introducing witnesses in the case of existential quantifiers. This familiar view will be used for automating the reasoning in CL, logging the instances of the axioms that have been applied, the cases that have been distinguished, and the witnesses that have been introduced. When a proof has been found, a natural deduction-style proof object (a typed lambda-term) can be reconstructed on the basis of this logbook.

The proof theory is complete and reasoning in CL is constructive in the sense of intuitionistic logic. Completeness proofs can be found in Bezem and Coquand [1, 2]. We have implemented the CL proof procedure in Prolog, see [3, CL.pl]. The implementation generates Coq proof scripts. This does not increase the size of the trusted core of Coq. Elaborated examples of the reasoning mechanism are given in Section 4 and in the Appendix.

3 Projective Plane Geometry

3.1 Axioms for Humans

In a projective plane there are *points* and *lines*, and there is one primitive relation between these, the *incidence* relation. Let uppercase letters range over points and lowercase letters over lines. If point P and line ℓ are incident, notation $P|\ell$, we say that 'P lies on ℓ' and that 'ℓ passes through P'. A point lying on two lines ℓ and m is called their *intersection*, and is written (ℓm). Dually, a line passing through two points P and Q is called their *connecting line*, and is written (PQ). A set of points is said to be *collinear* if there exists a line incident with all points in the set. Dually, a set of lines is said to be *concurrent* if there exists a point incident with all lines in the set.

We axiomatize the projective plane with the following three axioms.

Axiom 1 Any two points are incident with a line.

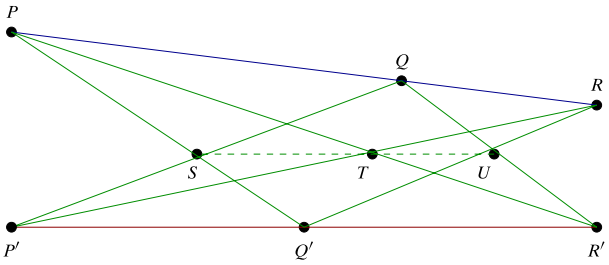
Dually, interchanging points and lines, we postulate:

Axiom 2 Any two lines are incident with a point.

The following self-dual axiom ensures that (PQ) and (ℓm) are uniquely determined for distinct P, Q and ℓ, m, respectively.

Axiom 3 Two distinct points cannot both be incident with two distinct lines.

Fig. 1 Pappus' axiom



There are some subtle differences with other, perhaps more familiar, formulations; we discuss the correspondence with two of the axioms given in Coxeter [6, p. 13]:

> Any two distinct points are incident with just one line.
> Any two distinct lines are incident with just one point.

These axioms are a fine example of the efficient use of natural language in informal classical mathematics. For the formalist and the constructivist they are horrendous. The *just one* quantification is an existential quantification containing an implicit universal quantification. The qualifier *distinct* introduces a negative condition; the two points should not be equal. The two statements $\neg\phi \Rightarrow \psi$ and $\phi \vee \psi$ are equivalent in classical logic, but constructively the latter is stronger than the former. Moreover, for atomic ϕ and ψ, $\phi \vee \psi$ is coherent and $\neg\phi \Rightarrow \psi$ is not.

It is not hard to see that the two above axioms are implied by Axioms 1–3. For the reverse implication, we need the existence of two distinct points, which follows from other axioms listed in Coxeter [6], as well as classical logic.

We now present Pappus' axiom;[1] see Fig. 1. Coxeter [6] states Pappus' axiom as follows:

> If alternate vertices of a hexagon lie on two lines, the three pairs of opposite sides meet in three collinear points.

Thus, given a hexagon $PQ'RP'QR'$, where the points P, Q, R are on one line and points P', Q', R' are on another line, the pairwise intersections

$$S \equiv ((PQ')(QP')) \qquad T \equiv ((RP')(PR')) \qquad U \equiv ((QR')(RQ'))$$

lie on the so-called *Pappus line*, the dashed line in Fig. 1. It is convenient to give a Pappus configuration by a 3×3 matrix (P_{ij}). The intersection of $(P_{1i}P_{2j})$ and $(P_{2i}P_{1j})$ is then P_{3k}, for all different i, j, k. This can be visualized by striking out the rows and columns in which P_{1i}, P_{2j} occur. For example, the matrix corresponding to the configuration in Fig. 1 reads

$$\begin{pmatrix} P & Q & R \\ P' & Q' & R' \\ U & T & S \end{pmatrix}.$$

[1] Pappus' axiom is often referred to as Pappus' *theorem*, because it is true in, for example, the real projective plane and in all finite projective planes. Pappus' axiom is, however, not true in all projective planes.

In order to exclude some degenerate cases in which the intersections S, T, U are indeterminate and possibly not collinear, Pappus' axiom requires some side-conditions, of which several other formulations (often not explicitly stated) can be found in the literature. We discuss some of these formulations and show how they relate to our formulation.

A. In the formulation of Coxeter [6], we believe these side-conditions are present in the use of the word 'hexagon'. The hexagon is assumed to be nondegenerate, that is, its sides are pairwise distinct.
B. A close reading of Cronheim [7] reveals the assumption of six *distinct* points, three on one line and three on a *distinct* line.
C. Another variation is to require that none of P, Q, R is incident with the line that joins P', Q', R', and vice versa.
D. We choose yet another formulation, which ensures determinacy of the intersections S, T, U by requiring the lines that should determine these intersections to be distinct.

It is easily seen that both A and B imply D. Concerning C, we formally verified the logical equivalence of the version of Pappus' axiom with side conditions C and the one with side conditions D in the presence of Axioms 1–3 ([19, GEO167,168]). The simple fact that C is expressed by a conjunction of length six, whereas D by one of length three, lengthens the proof search so that we prefer the latter.

Axiom 4 (Pappus) For collinear P, Q, R and collinear P', Q', R', the intersections $((QR')(RQ'))$, $((RP')(PR'))$, and $((PQ')(QP'))$ are collinear if they are determinate'.

As first observed by Hessenberg [11], Axioms 1–4 imply Desargues' axiom,[2] stated as follows:

Axiom 5 (Desargues) Two triangles perspective from a point are perspective from a line (under suitable side conditions).

Two triangles $A_1A_2A_3$ and $B_1B_2B_3$ are said to be perspective from a point S if the three lines joining corresponding vertices meet in S. Dually, two triangles are said to be perspective from a line ℓ if the three intersections of corresponding edges are joined by ℓ. S and ℓ are called the *perspectivity point* and *perspectivity line*, respectively. An example Desargues configuration is depicted in Fig. 2.

In the next subsection we present the 'machine' versions of Axioms 1–4. This 'mechanization' involves the unfolding of defined notions such as 'collinear', 'determinate', and 'perspective', in terms of the primitive incidence relation. Also, some formulas have to be replaced by (classical) equivalents in order to comply with the CL-format as defined in Section 1.

[2] Also this axiom is often referred to as a *theorem*, because it is true in, for example, the real projective plane. Unlike Pappus' axiom, Desargues' axiom is false in some finite projective planes.

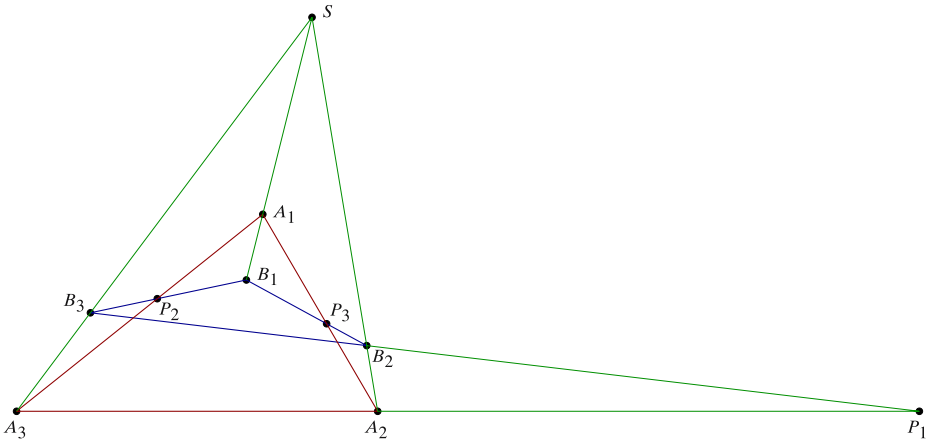


Fig. 2 Example of a Desargues configuration. In space, with the triangles $A_1A_2A_3$ and $B_1B_2B_3$ in different planes, the perspectivity line is the intersection of these two planes. This gives an easy proof for the two-dimensional case as well, essentially using the third dimension

3.2 Axioms for Machines

We formalize elementary projective plane geometry as a one-sorted theory. There exists a standard reduction of many-sorted logic to one-sorted. For every sort s one introduces a unary predicate $s(x)$ whose purpose is to express that x is of sort s. Quantifications with respect to sort s are then relativized. This means that everywhere $\forall_s x.\, \phi$ becomes $\forall x.\, (s(x) \Rightarrow \phi)$ and $\exists_s x.\, \phi$ becomes $\exists x.\, (s(x) \wedge \phi)$. The equality predicates for the various sorts are all replaced by one-sorted equality. Note that relativization essentially preserves the coherent format. In this way any many-sorted (coherent) formula ϕ can be translated into a one-sorted (coherent) formula ϕ'. We then have the following well-known result.

$$\models_m \phi \quad \text{if and only if} \quad \Delta \models \phi' \tag{$*$}$$

Here $\models_m$ expresses truth in all many-sorted models and Δ contains some axioms related to the translation $'$. It is standard to let Δ consist of axioms $\exists x.\, s(x)$ for all sorts s. These axioms are necessary because domains in FOL are nonempty, and they are also sufficient for obtaining the equivalence ($*$). It is less well-known that one may actually add several other axioms to Δ and still get equivalence. Of course the idea is not so much to complicate Δ unnecessarily as to take advantage of the extra axioms in simplifying ϕ'. We elaborate on this in the next paragraphs.

In order to simplify matters, we restrict attention to the case of plane geometry with a sort for points and a sort for lines, and we drop the convention of using upper case letters for points. Besides those two sorts the signature contains $|$ expressing incidence of a point with a line. The one-sorted signature would then consist of two extra unary predicates $\mathsf{point}(x)$ and $\mathsf{line}(y)$ besides $x \,|\, y$ expressing that 'object' x lies on 'object' y. Here the intended meaning of 'object' is point for x and line for y, but these meanings are not imposed by one-sorted $|$ but by neighboring atoms $\mathsf{point}(x)$ and $\mathsf{line}(y)$ coming from the relativized quantifiers.

A more informative $|$ is desirable. The axiom we would like to add to the standard Δ is $x|y \Rightarrow \mathsf{point}(x) \wedge \mathsf{line}(y)$. Why would such an extension be allowed? In order to see this, we must enter the standard argument for the equivalence $(*)$, in particular from right to left as strengthening Δ amounts to weakening the right-hand side. This argument is based on transforming any many-sorted model into a one-sorted model of Δ in the following way. Let the one-sorted domain be the union of the domains of the many-sorted model. Interpret the unary predicates $\mathsf{point}(x)$ and $\mathsf{line}(y)$ as the subsets of points and of lines, respectively, of this union. Interpret $x|y$ by the set of pairs of points and lines that are incident in the many-sorted model. The standard argument now proceeds by proving by formula induction that any ϕ is true in the many-sorted model if and only if ϕ is true in the corresponding one-sorted model of Δ. This argument can still be used. The only extra observation we make is that the one-sorted model also validates the axiom $x|y \Rightarrow \mathsf{point}(x) \wedge \mathsf{line}(y)$ added above. This completes the justification of the extension of Δ.

Thus we add $x|y \Rightarrow \mathsf{point}(x) \wedge \mathsf{line}(y)$ to the standard axioms:

$$\Delta = \{\exists x.\, \mathsf{point}(x), \exists x.\, \mathsf{line}(x), (x|y \Rightarrow \mathsf{point}(x) \wedge \mathsf{line}(y))\}$$

What then are the benefits of this extension? In order to answer this question, we observe that $\mathsf{point}(x) \wedge \phi \wedge x|y$ can be simplified to $\phi \wedge x|y$ and $\mathsf{line}(y) \wedge \phi \wedge x|y$ to $\phi \wedge x|y$. This allows us to economize 1, 1, 4 and 18(!) line- and point-atoms in the respective axioms below.

Axiom 1′ $\mathsf{point}(x) \wedge \mathsf{point}(y) \Rightarrow \exists u.\, (x|u \wedge y|u)$

Axiom 2′ $\mathsf{line}(u) \wedge \mathsf{line}(v) \Rightarrow \exists x.\, (x|u \wedge x|v)$

Axiom 3′ $x|u \wedge x|v \wedge y|u \wedge y|v \Rightarrow x = y \vee u = v$

Axiom 4′

$$\begin{aligned}
&x_1|u \wedge x_2|u \wedge x_3|u \wedge y_1|v \wedge y_2|v \wedge y_3|v\\
&\wedge x_1|\ell_1 \wedge y_2|\ell_1 \wedge p|\ell_1 \wedge x_2|\ell_2 \wedge y_1|\ell_2 \wedge p|\ell_2\\
&\wedge x_1|m_1 \wedge y_3|m_1 \wedge q|m_1 \wedge x_3|m_2 \wedge y_1|m_2 \wedge q|m_2\\
&\wedge x_2|n_1 \wedge y_3|n_1 \wedge r|n_1 \wedge x_3|n_2 \wedge y_2|n_2 \wedge r|n_2\\
&\Rightarrow \ell_1 = \ell_2 \vee m_1 = m_2 \vee n_1 = n_2 \vee \exists w.\, (p|w \wedge q|w \wedge r|w)
\end{aligned}$$

Axiom 4′ can be related to Fig. 1 by taking $x_1 = P, x_2 = Q, x_3 = R, \ldots, u = (PQ) = (QR)$, and so forth.

Axioms 1′–4′ are in CL-format and correspond to Axioms 1–4 of the previous section. Some remarks on logical equivalence are in order here. Note the positive formulation of Axiom 3′ as compared to Axiom 3. In Axiom 4′, collinearity of x_1, x_2, x_3, that is, $\exists u.\, (x_1|u \wedge x_2|u \wedge x_3|u)$, has been reformulated using the logical equivalence of $(\exists u.\, \phi(u)) \Rightarrow \psi$ and $\forall u.\, (\phi(u) \Rightarrow \psi)$ (u not free in ψ). Likewise for the collinearity of y_1, y_2, y_3. The condition enforcing the intersections p, q, r to be determinate, that is, $\ell_1 \neq \ell_2, m_1 \neq m_2$ and $n_1 \neq n_2$, has been moved to the conclusion using the logical equivalence of $(\neg\phi \wedge \psi) \Rightarrow \zeta$ and $\psi \Rightarrow (\phi \vee \zeta)$ (in classical logic).

The final step is the mechanization of Desargues' axiom 5. As this axiom is to be proved as a theorem, we may assume two triangles that are perspective from a point, satisfying certain side conditions. We then have to prove that there exists a

perspectivity line. The logical structure of this is extremely simple: a long list of facts (= closed atoms) and negated facts. Finally, the formula to be proved is

$$\exists \ell.\, (p_1 | \ell \wedge p_2 | \ell \wedge p_3 | \ell).$$

This is completely unproblematic from the point of view of CL, but geometrically the situation, in particular with respect to the side conditions, is so complicated that we prefer to explain this in a separate subsection.

3.3 Desargues Configurations

Definition 1 A *Desargues configuration* $\mathcal{D}$ is a sequence of points S, A_1, A_2, A_3, B_1, B_2, B_3, P_1, P_2, P_3 such that A_1, A_2, A_3 are distinct, B_1, B_2, B_3 are distinct, S, A_i, B_i are collinear for $i = 1, 2, 3$ and $(A_j A_k)$ and $(B_j B_k)$ are distinct lines meeting in P_i, for all rotations (i, j, k) of $(1, 2, 3)$.

Observe the following permutation invariance: if we have a Desargues configuration as above, then also S, A_i, A_j, A_k, B_i, B_j, B_k, P_i, P_j, P_k is a Desargues configuration, for any permutation (i, j, k) of $(1, 2, 3)$ (but only rotations will be used). For the purpose of convenient reference, we fix the names of these points, and we let $\mathcal{D}(x, y, z)$ denote the configuration obtained from permuting $(1, 2, 3)$ into (x, y, z). In particular we have $\mathcal{D} = \mathcal{D}(1, 2, 3)$.

Having an automated reasoning tool makes it attractive to experiment with different sets of side conditions. Cronheim's starting point for proving Desargues is a configuration consisting of seven *distinct* points (*non-collinear* A_1, A_2, A_3 and *non-collinear* B_1, B_2, B_3, and a point S) and three *distinct* lines $(A_i B_i)$ which meet in S, the perspectivity point. Note that we have allowed A_1, A_2, A_3 and/or B_1, B_2, B_3 to be collinear but require that corresponding 'edges of the triangles' be distinct.

Our set of conditions in Definition 1 is easily seen to follow from Cronheim's. Assume for example $(A_2 A_3) = (B_2 B_3)$. Then the points A_2 and B_2 lie on both $(A_2 B_2)$ and $(A_2 A_3) = (B_2 B_3)$. Hence by projective unicity (Axiom 3) the points A_2 and B_2 are equal or we have $(A_2 B_2) = (A_2 A_3)$. Similarly, A_3 and B_3 are equal or $(A_3 B_3) = (A_2 A_3)$. In all cases we violate Cronheim's conditions.

For instance, the configurations displayed in Fig. 3, satisfy our conditions but not Cronheim's. Nevertheless, there is a perspectivity line. It turned out that the weaker conditions as formulated in Definition 1 were sufficient for the proof.

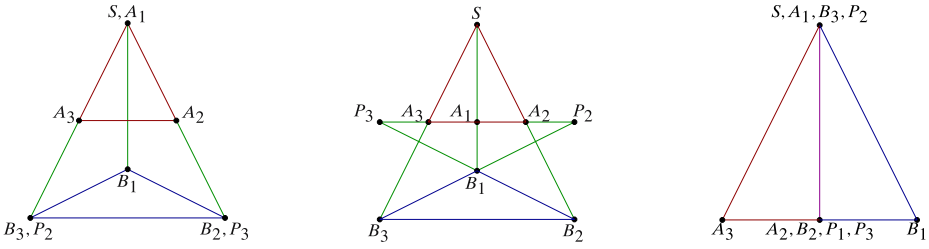


Fig. 3 Three degenerate Desargues configurations. *Left*: One of the vertices (A_1) is the point of perspectivity (S); lines $(A_2 A_3)$ and $(B_2 B_3) = (P_3 P_2)$ meet in P_1 at infinity. *Middle*: A degenerate triangle $A_1 A_2 A_3$, where $(A_2 A_3)$ and $(B_2 B_3)$ meet in P_1 at infinity. *Right*: Another degenerate case, where $(A_1 A_2)$ and $(B_2 B_3)$ are equal

It is very well possible that the proof can be carried out under even weaker conditions. We have not analyzed this any further. As a historical note we mention that leaving out the condition that A_1, A_2, A_3 are distinct and B_1, B_2, B_3 are distinct, such as done in Skolem [18, p. 129], leads to an axiom that implies that *any three points* are collinear. This actually provides an interesting example that is given in full detail in the Appendix.

Axiom 5 can now be reformulated as follows.

Axiom 6 (Desargues) For any Desargues configuration $\mathcal{D}$ such as in Definition 1, there exists a perspectivity line joining P_1, P_2 and P_3.

Hessenberg's theorem for humans states that Axioms 1–4 imply Axiom 6. For machines it reads as follows.[3]

Theorem 1 (Hessenberg) *In the theory consisting of Axioms* $1'$–$4'$, Δ *and equality axioms (reflexivity, symmetry, transitivity, and congruence with respect to the incidence relation), we have*

$$\begin{aligned}
&A_1|\ell_1 \wedge B_1|\ell_1 \wedge S|\ell_1 \wedge A_2|\ell_2 \wedge B_2|\ell_2 \wedge S|\ell_2 \wedge A_3|\ell_3 \wedge B_3|\ell_3 \wedge S|\ell_3\\
&\wedge A_2|a_1 \wedge A_3|a_1 \wedge P_1|a_1 \wedge A_3|a_2 \wedge A_1|a_2 \wedge P_2|a_2\\
&\wedge A_1|a_3 \wedge A_2|a_3 \wedge P_3|a_3 \wedge B_2|b_1 \wedge B_3|b_1 \wedge P_1|b_1\\
&\wedge B_3|b_2 \wedge B_1|b_2 \wedge P_2|b_2 \wedge B_1|b_3 \wedge B_2|b_3 \wedge P_3|b_3\\
&\Rightarrow A_1 = A_2 \vee A_2 = A_3 \vee A_3 = A_1 \vee B_1 = B_2 \vee B_2 = B_3 \vee B_3 = B_1\\
&\quad \vee a_1 = b_1 \vee a_2 = b_2 \vee a_3 = b_3 \vee \exists \ell.\, (P_1|\ell \wedge P_2|\ell \wedge P_3|\ell).
\end{aligned}$$

4 The Reasoning Mechanism

This section describes how the reasoning mechanism of CL works. Abstractly, this is by forward ground reasoning with case distinction to deal with disjunctions and introduction of new constants to deal with existential quantification.

As an example we prove that any projective plane with at least four points has at least three collinear points. Informally, the proof proceeds as follows. Consider the intersection Q of the line joining two points with the line joining two other points. If Q is different from the initial four points, then both lines have at least three points. Otherwise, if Q is equal to one of the initial four points, say, the first, then the line through the third and the fourth point has at least three points on it.

For a formal proof in CL, consider the theory consisting of Axioms $1'$, $2'$, Δ and equality axioms. Assume constants p_i and facts axiomatizing them as four different points: $\mathsf{point}(p_i)$ and $p_i \neq p_j \quad (1 \leq i < j \leq 4)$, which together form the (initial) *reasoning state*. The goal is to prove

$$\exists u\, x\, y\, z.\, (x|u \wedge y|u \wedge z|u \wedge x \neq y \wedge y \neq z \wedge x \neq z).$$

For this goal to be a CL formula, $x \neq y$ must be an atomic formula and cannot be taken as shorthand for $x = y \Rightarrow \bot$. This means that we have to define $\neq$ as the

[3] Here, in order to maintain the correspondence with Fig. 2, we do not adhere to our convention of using lowercase letters for points.

complement of $=$, which is done by extending the theory with the following two CL axioms (a so-called *definitional extension*).

$$x = y \vee x \neq y \qquad\qquad x = y \wedge x \neq y \Rightarrow \bot$$

In the rest of this section we reason in the extended theory.

Given a reasoning state, a *reasoning step* consists in, first, picking a closed instance $\boldsymbol{A} \Rightarrow \boldsymbol{D}$ of an axiom which is *invalid* in the state. This means that the antecedent $\boldsymbol{A}$ is true in the state, but the consequent $\boldsymbol{D}$ is not. More precisely, this means that all facts in $\boldsymbol{A}$ occur in the state, but for no disjunct $\exists \vec{x}.\, \boldsymbol{C}_j$ of $\boldsymbol{D}$ there exist witnesses $\vec{w}$ such that $\boldsymbol{C}_j[\vec{x}{:=}\vec{w}]$ is true in the state.

What happens then depends on the form of the conclusion $\boldsymbol{D}$.

- If $\boldsymbol{D}$ is a disjunction of length zero, that is, $\boldsymbol{D} = \bot$, then we are done, and any conclusion is valid.
- If $\boldsymbol{D}$ is a disjunction of length one without existential quantifiers, then $\boldsymbol{D}$ is a conjunction of facts, and we simply add these facts to the state and continue.
- If $\boldsymbol{D}$ is a disjunction of length one with existential quantifiers, $\boldsymbol{D} = \exists \vec{x}\, \boldsymbol{C}_1$, we introduce new constants $\vec{w}$ as witnesses and instantiate $\boldsymbol{C}_1$ with these constants, $\boldsymbol{C}_1[\vec{x}{:=}\vec{w}]$, add the facts to the state and continue. The state is understood to be extended also by the new constants, which from now on may be used in closed instances of axioms.

Before we continue with the case of a disjunction of length greater than one, let us illustrate the mechanism described so far by elaborating the example. In the initial state above we have the facts $\mathsf{point}(p_1)$ and $\mathsf{point}(p_2)$. The instance $\mathsf{point}(p_1) \wedge \mathsf{point}(p_2) \Rightarrow \exists u.\, (p_1 \,|\, u \wedge p_2 \,|\, u)$ of Axiom $1'$ is invalid since there is no line in the initial state joining p_1 and p_2. Applying this axiom 'remedies' this situation: we add a constant ℓ_{12} and facts $p_1 \,|\, \ell_{12}$ and $p_2 \,|\, \ell_{12}$ to the state. Note that the name 'ℓ_{12}' is irrelevant as long as it is new. Applying the same axiom, but now instantiated with p_3 and p_4, leads to the further extension of the state with a constant ℓ_{34} and facts $p_3 \,|\, \ell_{34}$ and $p_4 \,|\, \ell_{34}$. The instances $p_1 \,|\, \ell_{12} \Rightarrow \mathsf{point}(p_1) \wedge \mathsf{line}(\ell_{12})$ and $p_3 \,|\, \ell_{34} \Rightarrow \mathsf{point}(p_3) \wedge \mathsf{line}(\ell_{34})$ of the axiom $x \,|\, y \Rightarrow \mathsf{point}(x) \wedge \mathsf{line}(y)$ from Δ are invalid, so we add the facts $\mathsf{line}(\ell_{12})$ and $\mathsf{line}(\ell_{34})$ to the state (facts $\mathsf{point}(p_1)$ and $\mathsf{point}(p_3)$ are already present). Next, we consider the instance $\mathsf{line}(\ell_{12}) \wedge \mathsf{line}(\ell_{34}) \Rightarrow \exists x.\, (x \,|\, \ell_{12} \wedge x \,|\, \ell_{34})$ of Axiom $2'$. This instance is invalid: in the current state there exists no intersection of ℓ_{12} and ℓ_{34}. Therefore we introduce a constant q and add the facts $q \,|\, \ell_{12}$ and $q \,|\, \ell_{34}$ to the state. In a similar way as above the fact $\mathsf{point}(q)$ is added. Summing up, the reasoning state now extends the initial state with constants ℓ_{12}, ℓ_{34}, q and facts $\mathsf{line}(\ell_{12})$, $\mathsf{line}(\ell_{34})$, $p_1 \,|\, \ell_{12}$, $p_2 \,|\, \ell_{12}$, $p_3 \,|\, \ell_{34}$, $p_4 \,|\, \ell_{34}$, $\mathsf{point}(q)$, $q \,|\, \ell_{12}$, $q \,|\, \ell_{34}$.

We continue the description of the reasoning mechanism.

- If $\boldsymbol{D}$ is a disjunction of length greater than one, then the reasoning mechanism distinguishes as many cases as there are disjuncts in the disjunction. These cases are treated as disjunctions of length one as described above. In all these cases the goal has to be proved.

Let us continue the example with a disjunction of length two. In the state we reached above the instance $q = p_1 \vee q \neq p_1$ is invalid. The proof can now be completed by the following case distinction.

$q = p_1$ We add this fact to the state and infer $p_1 \mid \ell_{34}$ from $q \mid \ell_{34}$. Now that we have the facts $p_1 \mid \ell_{34}$, $p_3 \mid \ell_{34}$, $p_4 \mid \ell_{34}$, $p_1 \neq p_3$, $p_3 \neq p_4$, $p_1 \neq p_4$, the goal holds by taking ℓ_{34}, p_1, p_3, p_4 for u, x, y, z, respectively.

$q \neq p_1$ We add this fact to the state and apply the invalid instance $q = p_2 \vee q \neq p_2$, which gives rise to two subcases:

- **$q = p_2$** This subcase is dealt with analogously to the case $q = p_1$.
- **$q \neq p_2$** We add this fact to the state and have facts $q \neq p_1$, $p_1 \neq p_2$, $q \neq p_2$, $q \mid \ell_{12}$, $p_1 \mid \ell_{12}$, $p_2 \mid \ell_{12}$ so that again the goal holds.

5 The Complete Proof by Cronheim

Cronheim's proof [7] of Hessenberg's theorem is three pages long and has a remarkable level of detail. As can be guessed from the length of the machine proof (thousands of steps), there are nevertheless quite a few details left out, something which greatly improves the readability. However, in some cases, leaving out 'details' leads to incomplete or wrong proofs. This is what happened in Hessenberg's original argument. In some rare cases this may even lead to erroneous theorems.

In a formalization these details must all be taken care of, which is time consuming and tedious. It is here that we think that tools like the one used in this paper have something to offer. The CL prover turned out to be able to deal with all the details left out by Cronheim. Moreover, we were able to leave out many of the details that Cronheim deemed worth a few lines, mainly the justifications of application of Pappus' axiom.

Thus the proof *scripts* are considerably shorter than the original text. The ratio between the proof script describing (or rather generating) the formal proof and the original text in informal mathematics is usually called the *De Bruijn factor* (after N.G. de Bruijn, see [14]). In the early days of formalization the *De Bruijn factor* was around ten. Nowadays, it is around four. Here it is in total around one, and considerably smaller for some parts of the proof.

We give a high-level exhibition of the machine proof that we have constructed in the proof assistant Coq with the help of the CL prover. The proof closely follows Cronheim's proof. Minor modifications will be justified on the fly.

Cronheim distinguishes two cases: the general case, which is caught by Hessenberg's original argument, and a special case which was overlooked for 50 years. The case distinction can be phrased as either ϕ or $\neg\phi$, where ϕ abbreviates. "There exists a permutation (i, j, k) of $(1, 2, 3)$ such that $\neg A_i \mid (B_j B_k)$ and $\neg B_k \mid (A_i A_j)$". We reformulate $\neg\phi$ into CL by ψ.

$$\text{for all rotations } (i, j, k) \text{ of } (1, 2, 3),\ A_i \mid (B_j B_k) \text{ or } B_k \mid (A_i A_j) \qquad (\psi)$$

The switch from 'permutation' to 'rotation' will be justified in Subsection 5.2.

Observe that ψ amounts to 2^3 cases. In Subsection 5.2, still following Cronheim, we show that these can be reduced to 2 cases (Lemma 2): one triangle *circumscribes* the other, a notion defined as follows.

Definition 2 Triangle $B_1 B_2 B_3$ *circumscribes* triangle $A_1 A_2 A_3$ if for all rotations (i, j, k) of $(1, 2, 3)$ we have $A_i \mid (B_j B_k)$.

Figure 7 displays an example configuration where $B_1B_2B_3$ circumscribes $A_1A_2A_3$. The existence of a perspectivity line in this special setting is proved in Subsection 5.3 (Lemma 3). First, in Subsection 5.1, we treat the case of $\neg\psi$ as proved by Hessenberg's original argument (Lemma 1). Finally, in Subsection 5.4 we assemble these results to prove the main theorem (Theorem 1).

For the CL tool to construct the formal proof of Lemmas 1–3, only the essential steps had to be specified, namely the construction of the intersection of two given lines, the construction of a line through two given points and of a new line through *three* given points using Pappus' axiom. In the latter case it was never necessary to specify the two lines with each three points and the three pairs of lines whose respective intersections are collinear by the new line to be constructed. Neither was it necessary to give the details of the proof that the application of Pappus' axiom was justified. All files can be found on [3].

5.1 Hessenberg's Incomplete Argument

In this section we reproduce the argument which Hessenberg took for a complete proof. In fact this argument proves only the existence of a perspectivity line under some additional, nontrivial assumptions.

Lemma 1 *Let $\mathcal{D}$ be a Desargues configuration. Then there exists a perspectivity line joining P_1, P_2, P_3, or $A_1|(B_2B_3)$, or $B_3|(A_1A_2)$.*

Proof Consider Fig. 2 on page 67, and define four further points.

$$Q \equiv ((A_1A_2)(B_3B_2)) \qquad E \equiv ((A_1A_3)(SQ))$$
$$X \equiv ((A_1B_3)(SA_2)) \qquad F \equiv ((B_1B_3)(SQ))$$

Then, the points P_i are shown to be collinear by three applications of Axiom 4, as shown in Figs. 4, 5, and 6.

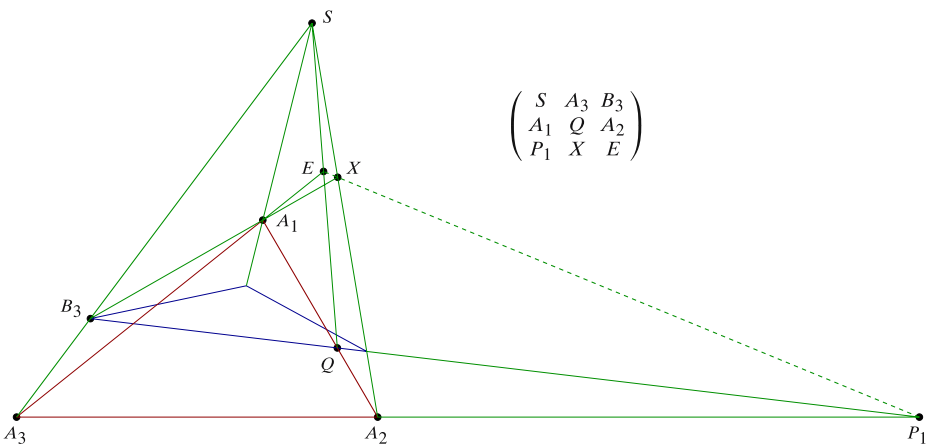


Fig. 4 Proof of Lemma 1: first application of Axiom 4

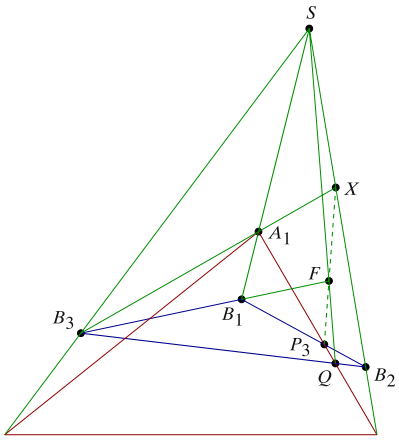


Fig. 5 Proof of Lemma 1: second application of Axiom 4

$$\begin{pmatrix} S & A_1 & B_1 \\ B_3 & B_2 & Q \\ P_3 & F & X \end{pmatrix}$$

The proof of this lemma can be generated automatically by using the following Prolog list as a proof script:

```
[meet(a1a2,b2b3,Q),join(a1,b3,A1B3),meet(s2,A1B3,X),join(s,Q,
SQ),meet(a3a1,SQ,E),meet(b3b1,SQ,F),pappus(p1,E,X),pappus(p3,
F,X),pappus(p1,p2,p3)]
```

In Prolog, identifiers starting with a capital are variables. The constant `a1` represents the point A_1 in the Desargues configuration, `a1a2` the line joining A_1 and A_2, and so on.

The terms in the above list control the application of their axioms. For example, `meet(L,M,P)` controls the application of the following Prolog clause representing[4] Axiom 2′:

```
meet(L,M,P)axiom meet(L,M):(l(L),l(M)=>dom(P),i(P,L),i(P,M)).
```

If the clause applies, `P` becomes bound to a new constant, representing the intersection of the lines `L` and `M`. The new constant is added to the domain by the presence of the atom `dom(P)` in the conclusion of the rule. Atoms like `dom(P)`, with `P` not occurring in the antecedent of the rule, represent existential quantification. Thus, for example, `meet(a1a2,b2b3,Q)` means the following: if `l(a1a2)` and `l(b2b3)`, then the intersection of the lines `a1a2` and `b2b3` is represented by a new constant to which variable `Q` is bound. Furthermore, the term `meet(L,M)` keeps track of which instance has been used, in order to be able to generate the proof. Terms `join(P,Q,L)` have an explanation dual to that of `meet(L,M,P)`. Terms `pappus(P,Q,R)` mean: prove that `P,Q,R` are collinear by an application of the Pappus' axiom. This axiom contains 14 other variables that are left unspecified, so that finding the right instance is nontrivial.

Thanks to the nine 'stepping stones' in the proof script above, which closely follows the proof by Cronheim, the formal proof is found in 901 applications of

[4]More information on the Prolog representation of axioms can be found in the Appendix.

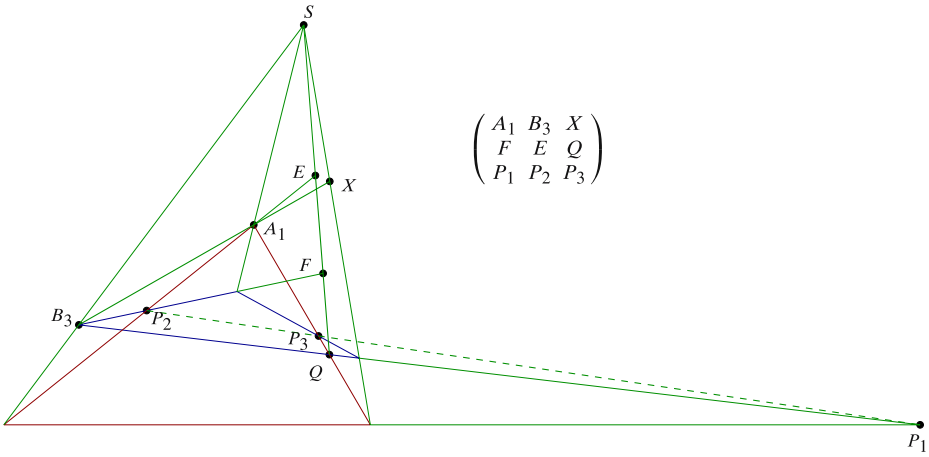


Fig. 6 Proof of Lemma 1: third application of Axiom 4

axioms, 365 of which actually are used in the proof. All files concerning this lemma can be found on [3, cro_case1.*].

The gap in Hessenberg's original proof was that $Q = F = B_3$ if $B_3|(A_1A_2)$ and $Q = E = A_1$ if $A_1|(B_2B_3)$. Then in particular the third application of Pappus' axiom (see Fig. 6), cannot be justified. Therefore the disjuncts $A_1|(B_2B_3)$ and $B_3|(A_1A_2)$ have been added to the conclusion of Lemma 1.

Corollary 1 *Let $\mathcal{D}$ be a Desargues configuration. Then there exists a perspectivity line joining P_1, P_2, P_3 or, for any permutation (i, j, k) of $(1, 2, 3)$, $A_i|(B_jB_k)$ or $B_k|(A_iA_j)$.*

5.2 Reducing 8 Gaps to 2

There is some redundancy in [7, p. 219 (2)] that we wish to avoid in our formalization. Let us first reformulate the premiss of [7, (2)] "There does not exist a permutation (i, j, k) of the numbers $(1, 2, 3)$ such that non(A_i, B_j, B_k) and non(B_k, A_i, A_j) simultaneously"[5] in a more positive way: for all permutations (i, j, k) one has $A_i|(B_jB_k)$ or $B_k|(A_iA_j)$. The conclusion of [7, (2)], "Either (A_x, B_y, B_z) for all permutations (x, y, z) or (B_x, A_y, A_z) for all permutations (x, y, z)", may be paraphrased as follows: One triangle circumscribes the other. This case is treated in Subsection 5.3.

There are six permutations of the numbers $(1, 2, 3)$. The even permutations correspond to rotations, the odd ones combine rotation with mirroring. In the premiss of [7, (2)] an odd permutation boils down to rotating *and interchanging* the two triangles. For example, the rotation $(3, 1, 2)$ yields the disjunction $A_3|(B_1B_2) \vee B_2|(A_3A_1)$. If we interchange the two triangles, we get $B_3|(A_1A_2) \vee A_2|(B_3B_1)$, which by commutativity corresponds with $(2, 1, 3)$, indeed an odd permutation. Since the conclusion of [7, (2)] is invariant under interchanging the two triangles, it can

[5] Cronheim's notation (P, Q, R) corresponds to $P|(QR)$ and 'non' stands for negation.

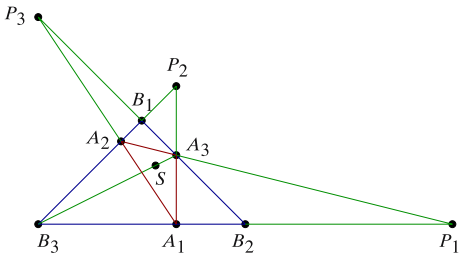


Fig. 7 $B_1B_2B_3$ circumscribes $A_1A_2A_3$

already be expected that one needs only permutations of one particular sign. This is indeed the case, and we prefer to restrict the premiss to the even permutations, that is, to the rotations. Under the premisses of Desargues' axiom, which are invariant under interchanging the two triangles, the system can automatically prove Lemma 2. As observed by Cronheim, this lemma is independent of Pappus' axiom.

Lemma 2 *If, for all rotations* (i, j, k) *of* $(1, 2, 3)$*, either* $A_i|(B_jB_k)$ *or* $B_k|(A_iA_j)$*, then there exists a perspectivity line, or triangle* $A_1A_2A_3$ *circumscribes triangle* $B_1B_2B_3$*, that is,* $A_i|(B_jB_k)$ *for all rotations* (i, j, k) *of* $(1, 2, 3)$*, or vice versa.*

This lemma can be proved fully automatically (also by some other theorem provers). It requires 1531 applications of axioms, 847 of which are used in the proof. Adding the disjunct 'there exists a perspectivity line' to the conclusion of the lemma above made it possible to prove Hessenberg's theorem with weaker side conditions than Cronheim's. All files concerning this lemma can be found on [3, cro_8_2.*].

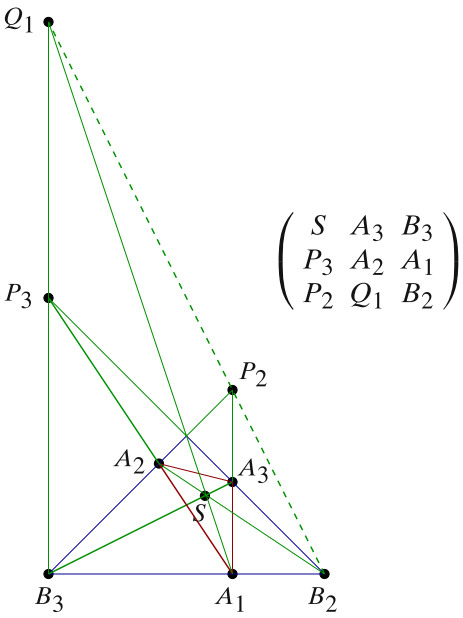


Fig. 8 Proof of Lemma 3: first application of Axiom 4

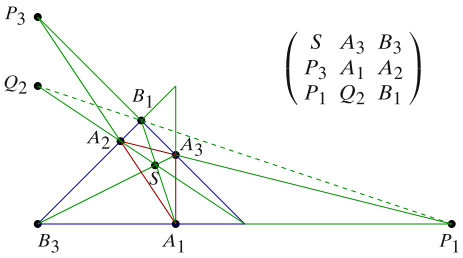


Fig. 9 Proof of Lemma 3: second application of Axiom 4

5.3 The Special Case: One Triangle Circumscribes the Other

An example of a configuration where a triangle $B_1B_2B_3$ circumscribes a triangle $A_1A_2A_3$ is depicted in Fig. 7. With S the point of perspectivity, this Desargues configuration is also known under the name of *Cevian triangles*.

Lemma 3 *For any Desargues configuration $\mathcal{D}$ where triangle $B_1B_2B_3$ circumscribes triangle $A_1A_2A_3$, there exists a perspectivity line joining the $P_i = ((A_jA_k)(B_jB_k))$.*

Proof Consider Fig. 7, and define the following points.

$$Q_1 \equiv ((B_3P_3)(SA_1)) \qquad Q_2 \equiv ((B_3P_3)(SA_2))$$

Then, collinearity of the P_i follows from three applications of Pappus (Axiom 4), as shown in Figs. 8, 9, and 10, respectively.

This lemma was proved automatically by the following proof script.

```
[join(b3,p3,B3P3),meet(B3P3,s1,Q1),meet(B3P3,s2,Q2),
pappus(p2,Q1,b2),pappus(p1,Q2,b1),pappus(p1,p2,p3)]
```

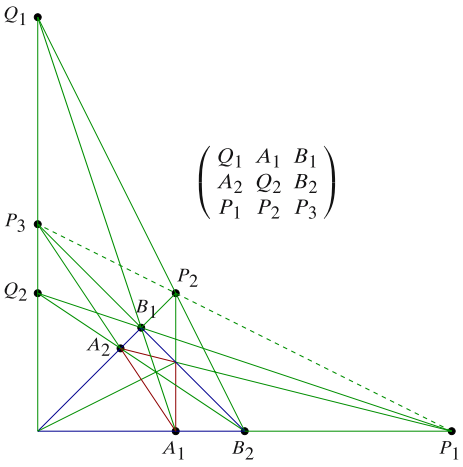


Fig. 10 Proof of Lemma 3: third application of Axiom 4

These six 'stepping stones' make it possible to find the formal proof in 1808 applications of axioms, 749 of which actually are used in the proof. All files concerning this lemma can be found on [3, cro_case2.*].

5.4 Assembling the Parts

We exhibit the proof of Hessenberg's theorem (Theorem 1) that we have formalized in the Coq proof assistant. The main Coq file [3, ht.v], written by hand, imports the modules [3, `cro_*.v`] that have been generated automatically by the CL prover. These files correspond to Lemmas 1–3, respectively. The proof of the theorem consists of three applications of Lemma 1, one of Lemma 2 and two of Lemma 3. In the main file, first the signature of points, lines, and incidence is declared, and the axioms of sorted incidence are listed. For equality we use Coq's built-in equality which is defined as the smallest reflexive relation, equivalent to Leibniz' equality. This equality is an equivalence relation, and congruence with respect to the incidence relation is easily proved. Then Axioms $1'$–$4'$ are listed.

To start the proof of the theorem, we consider an arbitrary Desargues configuration $\mathcal{D}$ (see Definition 1). Now, the goal, say γ, is to prove the existence of a perspectivity line joining the points P_i.

$$\gamma \equiv \exists \ell.\, (P_1 | \ell \wedge P_2 | \ell \wedge P_3 | \ell)$$

Applying Lemma 1 to a configuration $\mathcal{D}(x, y, z)$, we get that either γ, and then we are done, or $A_x | (B_y B_z) \vee B_z | (A_x A_y)$. Thus, by application of Lemma 1 to each of the rotations of $(1, 2, 3)$, we have asserted three disjunctions:

$$\begin{aligned} &A_1 | (B_2 B_3) \vee B_3 | (A_1 A_2), \\ &A_2 | (B_3 B_1) \vee B_1 | (A_2 A_3), \\ &A_3 | (B_1 B_2) \vee B_2 | (A_3 A_1). \end{aligned}$$

Given these disjunctions, we are ready to apply Lemma 2, by which we get two symmetrical cases, either $B_1 B_2 B_3$ circumscribes $A_1 A_2 A_3$, or vice versa. Both cases are solved by application of Lemma 3; for the second case the roles of A and B in Lemma 3 have to be interchanged.

6 Related Work and Future Research

Most theorem provers participating in the CADE ATP System Competition (CASC, see [20]) can prove Lemma 2, which is problem GEO169 in the TPTP database [19]. But no CASC system on TPTP is currently able to prove Lemma 1 (GEO166) or Lemma 3 (GEO165). Hessenberg's theorem (GEO164) is certainly out of reach for any of these systems. It can be expected that those systems can prove Lemmas 1 and 3 when given some hints as we did in CL. One problem with these fully automatic systems is that they are not designed for interactive use. More importantly, they do not generate portable, reusable proof objects in a standardized form.

Although well implemented, most systems in CASC Skolemize the input. For example, the Skolemized form of Axiom $1'$ would be

$$\mathsf{point}(x) \wedge \mathsf{point}(y) \Rightarrow x | f(x, y) \wedge y | f(x, y).$$

Here f is a Skolem function. However, $f(x, y)$ and $f(y, x)$ both represent the line through points x and y (even if there exists such a line already). The same is true for intersection, which leads to major inefficiencies that are avoided in CL. As yet it is not clear how important these inefficiencies are in fully automated theorem proving. There is only one system participating in CASC, Geo by de Nivelle and Meng [9], which is fully based on CL. Geo2006i ended in the sixth place of 11 systems competing in the category FOF (and got the prize for the best newcomer). Geo2007j did less well. Although we consider these results as promising, we envision that CL will be most useful for interactive theorem proving.

The translation of FOL to CL is not nearly as well developed as the translation of FOL to, for example, conjunctive normal form. The latter translation is treated in two chapters of [17], covering almost 100 pages. There exist two known translations of FOL to CL, one from Bezem and Coquand [2, Sec. 7] and one from de Nivelle and Meng [9]. They have not be compared in detail. Skolemization is avoided by both, and both are linear. As a consequence, the distance between the original FOL formula and its CL translation is much smaller than in the case of conjunctive normal form. In the case of Bezem and Coquand [2] this can be made precise in the following way: every FOL theory has a conservative extension that is equivalent to a CL theory, a result that goes back to Skolem [18]. It is important to note that this is a result in classical logic. Although reasoning in CL is constructive, translating, for example, $\neg\phi \Rightarrow \psi$ to $\phi \vee \psi$, is not. CL does not differ from resolution logic [17, Ch. 2] in this respect. The optimal translation of FOL to CL is clearly an important topic of future research.

Another challenge for CL is the integration in interactive theorem provers. In the current situation, exporting the (would-be) tautology, including the translation to CL-format, is done by hand. The CL prover generates a self-contained Coq proof script, which, in turn, is used by the Coq system to build a proof object. The generated file can be compiled and easily included in further Coq developments. It is desirable to have back-ends for other proof assistants, such as Isabelle [15], as well. It is also desirable to automate the exportation of formulas, or even whole proof states, and their conversion to CL-format.

Proof assistants such as Coq and Isabelle do already have tactics for automated reasoning. One can distinguish between general-purpose and special-purpose ones. General-purpose tactics are, for example, `tauto`, `first-order`, and `zenon` [5] in Coq and `blast` [16] in Isabelle. A strong point of these tactics is their smooth integration in the proof assistant. Some tactics have even higher-order reasoning capabilities, typically based on higher-order matching. It seems that `blast` performs best, followed by `zenon`. Both `blast` and `zenon` are based on the tableau method [17, Ch. 3]. There exists no systematic comparison of the performance of the various approaches in different proof assistants. A competition like CASC but then for systems generating standardized proof objects would make such a comparison possible.

Special-purpose tactics, such as for Presburger arithmetic or for equational reasoning in rings, can be quite powerful in their domain, but are not generic.

In Kusak and Leończuk [13] the formal verification of Hessenberg's theorem has been carried out *by hand* in Mizar, resulting in a proof of 15 pages. Mizar is based on FOL and has a structured taxonomy of theories. Although Mizar code is fully formal, it can be read like informal mathematics, but the intuitions are difficult to

grasp by the level of detail. In Kusak and Leończuk [13] there is no reference to an existing proof in the literature, but our impression is that the proof is different from Cronheim's. The proof has been verified automatically by Mizar, but no attempt has been made to generate parts of it. The Mizar proof format is not standardized and has not been ported to other systems.

7 Conclusion

We have argued in favor of the use of CL for the automation of interactive proof construction. We have illustrated our approach with a case study on Hessenberg's theorem. The choice of this case was deliberate: elementary projective geometry can be expressed completely within CL. This is not the case for all of FOL. However, the translation of full FOL to CL is much easier than to conjunctive normal form, as it does not involve Skolemization. This places CL somewhere in the middle between resolution logic (using conjunctive normal forms) and tableau methods (using general formulas). Resolution logic is the fastest known technique for automated reasoning in FOL, outperforming tableau methods by a wide margin. We expect the reasoning power of CL to lie somewhere between the power of resolution and that of tableau methods. The advantage CL has over resolution is the smaller distance between automated and interactive theorem proving, so that their integration will be easier. How to integrate and which interface to use to steer the CL prover will be subject to further research.

Appendix: Desargues' Axiom by Skolem

In Fig. 11 we have reproduced the example from Skolem [18, p. 29] in which Skolem illustrates the proof theoretic techniques developed earlier in his paper by showing that Desargues' axiom is independent of the basic axioms of projective plane

Fig. 11 Fragment from Skolem [18, p. 29]

Beispiel: Es sei zu untersuchen, ob der Desarguesche Satz von den homologen Dreiecken aus den aufgestellten Verknüpfungsaxiomen folge oder nicht. Dieser Satz ist ja ein deskriptiver. Er sagt in der kombinatorischen Sprache folgendes: Wenn die Paare

$$(A_1b_1)(A_1c_1)(B_1a_1)(B_1c_1)(C_1a_1)(C_1b_1)$$
$$(A_2b_2)(A_2c_2)(B_2a_2)(B_2c_2)(C_2a_2)(C_2b_2)$$
$$(A_1d)(A_2d)(Pd)(B_1e)(B_2e)(Pe)(C_1f)(C_2f)(Pf)$$
$$(Da_1)(Da_2)(Dp)(Eb_1)(Eb_2)(Ep)(Fc_1)(Fc_2)$$

vorkommen, dann soll auch mindenstens eines der Paare

$$(Fp)(a_1a_2)(b_1b_2)(c_1c_2)$$

vorhanden sein.

geometry. First Skolem formulates Desargues' axiom as a coherent formula. Instead of using an existential quantifier to express the collinearity of D, E and F he states that any line p joining D and E contains F. Since there is always such a line p, these two formulations are equivalent. More interesting are Skolem's side conditions, which appear positively in his disjunctive conclusion $(Fp)(a_1a_2)(b_1b_2)(c_1c_2)$. These side conditions have the same function as the side conditions in Pappus' axiom, namely, to cover the cases in which intersections would become indeterminate. It turns out that Skolem's side conditions are too weak and that therefore his formulation of Desargues' axiom is too strong. It allows us in fact to prove that any three points are collinear, thus trivializing the projective plane. As this proof is small, it provides a good example to demonstrate our machinery in full detail. By the way, Skolem's proof-theoretic argument applies equally well to a correct formulation of Desargues' axiom, for example, with a disjunctive conclusion

$$(Fp)(a_1a_2)(b_1b_2)(c_1c_2)(A_1B_1)(B_1C_1)(A_1C_1)(A_2B_2)(B_2C_2)(A_2C_2)$$

We stress that we chose this example as an historical anecdote that serves our explanatory purposes well and that we do not in any way intend to question the value of Skolem's contribution.

The automated reasoning tool [3, CL.pl] has been implemented in the programming language Prolog. In the input, file below, most of the clauses have the form `<tag> axiom <term> : (<formula>).` We explain each of the constituents:

`<tag>`, if different from '`_`', controls the use of the axiom. This is used only for one axiom, tagged `abc(P,Q)`. In combination with the `enabled` and the `next` predicates in the last two clauses of the input, this limits the construction of new lines to lines through *different* points `a`, `b`, `c`.
`<term>` gives a name to the axiom including all universally quantified variables. This is used to keep track of which instances of which axioms have been used.
`<formula>` states the coherent formula in question. Here Prolog syntax is used; that is, variables start with a capital, '`,`' stands for conjunction, '`=>`' for implication, '`;`' for disjunction. Finally, `dom` on the right of '`=>`' in the axioms for projective lines and points stands for existential quantification. If the axiom `line(P,Q)` is used, variable `L` is substituted by a fresh object (name), which is subsequently added to the domain. Dually for `point(L,M)`.

With this explanation, we trust that the comments after the symbol '%' sufficiently explain the file [3, sd.in] listed below.

```
name('DbyS').                                  % Desargues' axiom by Skolem
:- dynamic p/1,l/1,i/2,e/2.              % predicates p for point, l for line
                                         %    i for incidence, e for equality

dom(a). dom(b). dom(c).                        % constants a,b,c in the domain

_ axiom points : (true => p(a),p(b),p(c)).              % a,b,c are points

% goal is proved if a,b,c are collinear
_ axiom goal_proved(L) : (i(a,L),i(b,L),i(c,L) => goal).

_ axiom sortp(P,L) : (i(P,L) => p(P)).      % inc. pairs have points left
_ axiom sortl(P,L) : (i(P,L) => l(L)).      % and lines right
```

```
% equality axioms
_ axiom p_ref(X)    : (p(X) => e(X,X)).                  % refl. for points
_ axiom l_ref(X)    : (l(X) => e(X,X)).                  % refl. for lines
_ axiom sym(X,Y)    : (e(X,Y) => e(Y,X)).                % symmetry
_ axiom tra(X,Y,Z) : (e(X,Y),e(Y,Z) => e(X,Z)).          % transitivity

% congruence axioms
% equal points lie on the same lines
_ axiom conp(P,Q,L) : (e(P,Q),i(Q,L) => i(P,L)).
% equal lines have the same points
_ axiom conl(P,L,M) : (i(P,L),e(L,M) => i(P,M)).

% projective geometry (Axioms 1'-3')
abc(P,Q) axiom line(P,Q) : (p(P),p(Q) => dom(L),i(P,L),i(Q,L)).
_ axiom point(L,M) : (l(L),l(M) => dom(P),i(P,L),i(P,M)).
_ axiom uniq(P,Q,L,M) : (i(P,L),i(P,M),i(Q,L),i(Q,M) => e(P,Q);e(L,M)).

% Desargues' axiom as formulated by Skolem
% capital letters L prefix Skolem's names for lines in order to comply
% with Prolog's convention on variables
_ axiom wrong(A1,B1,C1,A2,B2,C2,La1,Lb1,Lc1,La2,Lb2,Lc2,
              P,Ld,Le,Lf,D,E,F,Lp):
(
  i(A1,Lb1),i(A1,Lc1),i(B1,La1),i(B1,Lc1),i(C1,La1),i(C1,Lb1),  % A1B1C1
  i(A2,Lb2),i(A2,Lc2),i(B2,La2),i(B2,Lc2),i(C2,La2),i(C2,Lb2),  % A2B2C2
  i(A1,Ld),i(A2,Ld),i(P,Ld), % \
  i(B1,Le),i(B2,Le),i(P,Le), % - P is the point of perspectivity
  i(C1,Lf),i(C2,Lf),i(P,Lf), % /
  % line Lp is the candidate perspectivity line through D and E
  i(D,La1),i(D,La2),i(D,Lp),i(E,Lb1),i(E,Lb2),i(E,Lp),i(F,Lc1),i(F,Lc2)
  =>
  % on which F should lie as well, or two corresping edges coincide
  i(F,Lp);e(La1,La2);e(Lb1,Lb2);e(Lc1,Lc2)
).

enabled(abc(P,Q),[]) :- member(P,[a,b,c]),member(Q,[a,b,c]),P \= Q.
next(abc(P,Q),[],[]).
```

Next we list the output file [3, sd.out],[6] which is self-explaining to a large degree. The only difficult point is the application of Skolem's formulation of Desargues' axiom, which we will explain at the end.

```
By axiom points using true we have:
  p(a) /\ p(b) /\ p(c)
By axiom p_ref(a) using p(a) we have:
  e(a,a)
By axiom p_ref(b) using p(b) we have:
  e(b,b)
By axiom p_ref(c) using p(c) we have:
  e(c,c)
By axiom line(a,b) using p(a) /\ p(b) we have:
  i(a,w0) /\ i(b,w0)
By axiom sortl(a,w0) using i(a,w0) we have:
  l(w0)
By axiom l_ref(w0) using l(w0) we have:
  e(w0,w0)
```

[6] Not to be confused with the Coq file [3, sd.v] the CL prover also generates.

```
By axiom line(a,c) using p(a) /\ p(c) we have:
  i(a,w1) /\ i(c,w1)
By axiom sortl(a,w1) using i(a,w1) we have:
  l(w1)
By axiom l_ref(w1) using l(w1) we have:
  e(w1,w1)
By axiom line(b,c) using p(b) /\ p(c) we have:
  i(b,w2) /\ i(c,w2)
By axiom sortl(b,w2) using i(b,w2) we have:
  l(w2)
By axiom l_ref(w2) using l(w2) we have:
  e(w2,w2)
By axiom wrong(a,a,a,c,c,a,w0,w0,w0,w1,w1,w2,a,w1,w1,w0,a,a,b,w1)
  using
  i(a,w0) /\ i(a,w0) /\ i(a,w0) /\ i(a,w0) /\ i(a,w0) /\ i(a,w0) /\
  i(c,w1) /\ i(c,w2) /\ i(c,w1) /\ i(c,w2) /\ i(a,w1) /\ i(a,w1) /\
  i(a,w1) /\ i(c,w1) /\ i(a,w1) /\ i(a,w1) /\ i(c,w1) /\ i(a,w1) /\
  i(a,w0) /\ i(a,w0) /\ i(a,w0) /\ i(a,w0) /\ i(a,w1) /\ i(a,w1) /\
  i(a,w0) /\ i(a,w1) /\ i(a,w1) /\ i(b,w0) /\ i(b,w2)
we have:
  i(b,w1) \/ e(w0,w1) \/ e(w0,w1) \/ e(w0,w2)
stack pushed, stack: i(b,w1) \/ e(w0,w1) \/ e(w0,w1) \/ e(w0,w2) :: nil


stack top tailed: i(b,w1)
By axiom goal_proved(w1) using i(a,w1) /\ i(b,w1) /\ i(c,w1) we have:
goal

valid, stack: e(w0,w1) \/ e(w0,w1) \/ e(w0,w2) :: nil

stack top tailed: e(w0,w1)
By axiom sym(w0,w1) using e(w0,w1) we have:
  e(w1,w0)
By axiom conl(b,w0,w1) using i(b,w0) /\ e(w0,w1) we have:
  i(b,w1)
By axiom goal_proved(w1) using i(a,w1) /\ i(b,w1) /\ i(c,w1) we have:
  goal

valid, stack: e(w0,w1) \/ e(w0,w2) :: nil

stack top tailed: e(w0,w1)
By axiom sym(w0,w1) using e(w0,w1) we have:
  e(w1,w0)
By axiom conl(b,w0,w1) using i(b,w0) /\ e(w0,w1) we have:
  i(b,w1)
By axiom goal_proved(w1) using i(a,w1) /\ i(b,w1) /\ i(c,w1) we have:
  goal

valid, stack: e(w0,w2) :: nil

stack popped: e(w0,w2)
By axiom sym(w0,w2) using e(w0,w2) we have:
  e(w2,w0)
By axiom conl(a,w0,w2) using i(a,w0) /\ e(w0,w2) we have:
  i(a,w2)
By axiom goal_proved(w2) using i(a,w2) /\ i(b,w2) /\ i(c,w2) we have:
  goal
valid, stack: nil

Yes
```

By matching the `wrong`-terms in input and output:

```
wrong(A1,B1,C1,A2,B2,C2,La1,Lb1,Lc1,La2,Lb2,Lc2,P, Ld, Le, Lf,D,E,F,Lp)

wrong( a, a, a, c, c, a, w0, w0, w0, w1, w1, w2,a, w1, w1, w0,a,a,b,w1)
```

we find that Desargues' axiom in Skolem's formulation is applied by the machine in the following completely degenerated case. The first triangle is the point `a`, the second triangle consists of the points `c`, `a`, and `a` is the point of perspectivity. The edges of the first triangle all coincide with the line `w0` joining `a`, `b`. The edges of the second triangle are `w1` $= a_2, b_2$ joining `a` $= C_2$, `c` $= A_2 = B_2$, as well as `w2` $= c_2$ joining `b`, `c` $= A_2 = B_2$. With this particular choice of edges, corresponding edges meet in `a`, `a`, `b`, respectively. Any line through `a` connects `D=a` and `E=a`, in particular `w1`. The conclusion that `F=b` lies on `w1` leads to the collinearity of `a`, `b`, `c`, and so does each of the other disjuncts in the conclusion: `i(b,w1)\/e(w0,w1)\/e(w0,w1)\/e(w0,w2)`.

References

1. Bezem, M., Coquand, T.: Newman's lemma—a case study in proof automation and geometric logic. In: Păun, G., Rozenberg, G., Salomaa, A. (eds.) Current Trends in Theoretical Computer Science, vol. 2, pp. 267–282. World Scientific, Singapore (2004)
2. Bezem, M., Coquand, T.: Automating coherent logic. In: Sutcliffe, G., Voronkov, A. (eds.) Proceedings LPAR-12. Lecture Notes in Computer Science, vol. 3825, pp. 246–260. Springer-Verlag, Berlin (2005)
3. Bezem, M., Hendriks, D.: Web page including CL tool, input files, Coq files. http://www.cs.vu.nl/~diem/research/ht/ .
4. Bezem, M., Hendriks, D., de Nivelle, H.: Automated proof construction in type theory using resolution. J. Autom. Reason. **29**(3), 253–275 (2002)
5. Bonichon, R., Delahaye, D., Doligez, D.: Zenon: an extensible automated theorem prover producing checkable proofs. In: Dershowitz, N., Voronkov, A. (eds.) Proceedings LPAR-14. Lecture Notes in Computer Science, vol. 4790, pp. 151–165. Springer-Verlag, Berlin (2007)
6. Coxeter, H.S.M.: The Real Projective Plane, 2nd edn. Cambridge University Press, Cambridge (1955)
7. Cronheim, A.: A proof of Hessenberg's theorem. Proc. AMS **4**(2), 219–221 (1953)
8. de Nivelle, H.: Translation of resolution proofs into short first-order proofs without choice axioms. Inf. Comput. **199**(1), 24–54 (2005) [Special Issue on the 19th International Conference on Automated Deduction (CADE-19)]
9. de Nivelle, H., Meng, J.: Geometric resolution: a proof procedure based on finite model search. In: Harrison, J., Furbach, U., Shankar, N. (eds.) International Joint Conference on Automated Reasoning 2006, p. 15. Springer, Seattle (2006)
10. Harper, R., Honsell, F., Plotkin, G.D.: A framework for defining logics. J. ACM, **40**(1), 143–184 (1993)
11. Hessenberg, G.: Beweis des Desarguesschen Satzes aus dem Pascalschen. Math. Ann. **61**, 161–172 (1905)
12. Johnstone, P.: Sketches of an Elephant: A Topos Theory Compendium, vol. 2. Oxford University Press, London (2002)
13. Kusak, E., Leończuk, W.: Hessenberg theorem. Formaliz. Math. **2**(2), 217–219 (1991)
14. Nederpelt, R.P., Geuvers, J.H., R.C. de Vrijer (eds.): Selected Papers on Automath. North-Holland, Amsterdam (1994)
15. Paulson, L.C.: Isabelle: A Generic Theorem Prover (with a contribution by T. Nipkow). Lecture Notes in Computer Science, vol. 828. Springer, Berlin (1994)
16. Paulson, L.C.: A generic tableau prover and its integration with Isabelle. J. Univers. Comput. Sci. **5**(3), 73–87 (1999)

17. Robinson, J.A., Voronkov, A. (eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press, Amsterdam (2001)
18. Skolem, Th.: Logisch-Kombinatorische Untersuchungen über die Erfüllbarkeit und Beweisbarkeit Mathematischen Sätze nebst einem Theoreme über Dichte Mengen, Skrifter I, **4**, 1–36, Det Norske Videnskaps-Akademi, 1920. In: Fenstad, J.E. (ed.) Selected Works in Logic, pp. 103–136. Universitetsforlaget, Oslo (1970)
19. Sutcliffe, G., Suttner, C.: The TPTP problem library: CNF release v1.2.1. J. Autom. Reason. **21**(2), 177–203 (1998)
20. Sutcliffe, G., Suttner, C.: The state of CASC. AI Commun. **19**(1), 35–48 (2006)
21. The Coq development team. The Coq Proof Assistant Reference Manual, version 8.1, 2006. http://coq.inria.fr/.