OTTER proofs in Tarskian geometry

Michael Beeson^1 and Larry Wos^2

^{1} San José State University
^{2} Argonne National Laboratory

Abstract. We report on a project to use OTTER to find proofs of the theorems in Tarskian geometry proved in Szmielew's part (Part I) of [9]. These theorems start with fundamental properties of betweenness, and end with the development of geometric definitions of addition and multiplication that permit the representation of models of geometry as planes over Euclidean fields, or over real-closed fields in the case of full continuity. They include the four challenge problems left unsolved by Quaife, who two decades ago found some OTTER proofs in Tarskian geometry (solving challenges issued in [15]).
Quaife's four challenge problems were: every line segment has a midpoint; every segment is the base of some isosceles triangle; the outer Pasch axiom (assuming inner Pasch as an axiom); and the first outer connectivity property of betweenness. These are to be proved without any parallel axiom and without even line-circle continuity. These are difficult theorems, the first proofs of which were the heart of Gupta's Ph. D. thesis under Tarski. OTTER proved them all in 2012. Our success, we argue, is due to improvements in techniques of automated deduction, rather than to increases in computer speed and memory.
The theory of Hilbert (1899) can be translated into Tarski's language, interpreting lines as pairs of distinct points, and angles as ordered triples of non-collinear points. Under this interpretation, the axioms of Hilbert either occur among, or are easily deduced from, theorems in the first 11 (of 16) chapters of Szmielew. We have found Otter proofs of all of Hilbert's axioms from Tarski's axioms (i.e. through Satz 11.49 of Szmielew, plus Satz 12.11). Narboux and Braun have recently checked these same proofs in Coq.

1 Introduction

Geometry has been a test bed for automated deduction almost as long as computers have existed; the first experiments were done in the 1950s. In the nineteenth century, geometry was the test bed for the development of the axiomatic method in mathematics, spurred on by the efforts to prove Euclid's parallel postulate from his other postulates and ultimately the development of non-Euclidean geometry. This effort culminated in Hilbert's seminal 1899 book [5]. In the period 1927–1965, Tarski developed his simple and short axiom system (described below). Some 35 years ago, the second author experimented with finding proofs from Tarski's axioms, reporting success with simple theorems, but leaving several unsolved challenge problems. The subject was revisited by Art Quaife, who in his 1992 book [8] reported on the successful solution of some of those challenge problems using an early version of McCune's theorem-prover, OTTER. But there were several theorems that Quaife was not able to get OTTER to prove, and he stated them as "challenge problems" in his book. As far as we know, nobody took up the subject again until 2012, when the present authors set out to see whether automated reasoning techniques, and/or computer hardware, had improved enough to let us progress beyond Quaife's achievements.

The immediate stimulus was the existence of the almost-formal development of many theorems in Tarskian geometry in Part I of [9]. This Part I is essentially the manuscript developed by Wanda Szmielew for her 1965 Berkeley lectures on the foundations of geometry, with "inessential modifications" by Schwäbhauser. There are 16 chapters. Quaife's challenge problems occur in the first nine chapters. The rest contain other important geometrical theorems (described below). We set ourselves the goal to find OTTER proofs of each of the theorems in Szmielew's 16 chapters. Our methodology was to make a separate problem of each theorem, supplying OTTER with the axioms and the previously-proved theorems, as well as the (negated) goal expressing the theorem to be proved. We were sometimes forced to supply OTTER with more information than that, as we describe below.

We did succeed in 2012 to find OTTER proofs of Quaife's challenge problems, the last of which is Satz 9.6 in Szmielew. Verification of chapters 1 through 11 (which we completed with OTTER in 2013) suffices to formally prove Hilbert's axioms from those of Tarski. In this paper, we give Tarski's axioms, explain the challenge problems of Quaife and some of the axioms of Hilbert, discuss the difficulties of finding OTTER proofs of these theorems, and explain what techniques we used to find those proofs.

2 Tarski's axioms

In about 1927, Tarski first lectured on his axiom system for geometry, which was an improvement on Hilbert's 1899 axioms in several ways: first, the language had only one sort of variables (for points), instead of having three primitive notions (point, line, and angle). Second, it was a first-order theory (Hilbert's axioms mentioned sets, though not in an essential way). Third, the axioms were short, elegant, and few in number (only about fifteen). They could be expressed comprehensibly in the primitive syntax, without abbreviations.

2.1 History

The development of Tarski's theory, started in 1927 or before, was delayed, first by Tarski's involvement in other projects, and then by World War II (galley proofs of Tarski's article about it were destroyed by bombs). The first publication of Tarski's axioms came in 1948 [10], and contained little more than a list of the axioms and a statement of the important metamathematical theorems about the theory (completeness, representation of models as $\mathbb{F}^2$ for $\mathbb{F}$ a real-closed field, quantifier-elimination and decidability). Tarksi then lectured on the subject at Berkeley in 1956–58, and published a reduced set of axioms in 1959 [11]. In the sixties, Tarski, Szmielew, Gupta, and Schwäbhauser (and some students) reduced the number of axioms still further. The manuscript that Szmielew prepared for her 1965 course became Part I of [9]. More details of the history of these axioms can be found in [12] (our main source) and the foreword to (the Ishi Press edition of) [9]. For our present purposes, the relevance of the history is mainly that there are three versions of Tarski's theory, the 1948 version, the 1959 version, and the 1965 version (published in 1983). The earlier experiments of Wos used the 1959 axioms, but Quaife used the 1965 version, as we do. The exact differences are explained below.

2.2 Syntax

The fundamental relations in the theory (first introduced by Pasch in 1852) are "betweenness", which we here write $\mathbf{T}(a,b,c)$, and "equidistance", or "segment congruence", which is officially written $E(a,b,c,d)$, and unofficially as $ab = cd$, segment ab is congruent to segment cd. The intuitive meaning of $\mathbf{T}(a,b,c)$ is that b lies between a and c on the line connecting a and c; Tarski used non-strict betweenness, so we do have $\mathbf{T}(a,a,c)$ and $\mathbf{T}(a,c,c)$ and even $\mathbf{T}(a,a,a)$. Hilbert used strict betweenness. Both of them wrote $\mathbf{B}(a,b,c)$, which is a potential source of confusion. We therefore reserve $\mathbf{B}$ for strict betweenness, and use $\mathbf{T}$ for Tarski's non-strict betweenness. The fact that Tarski's initial is 'T' should serve as a mnemonic device. Of course the equality relation between points is also part of the language.

2.3 Betweenness and congruence axioms

We write $ab = cd$ instead of $E(a,b,c,d)$ to enhance human readability. In OTTER files of course we use $E(a,b,c,d)$. The following are five axioms from the 1965 system.

$ab = ba$	(A1) reflexivity for equidistance
$ab = pq \wedge ab = rs \;\rightarrow\; pq = rs$	(A2) transitivity for equidistance
$ab = cc \;\rightarrow\; a = b$	(A3) identity for equidistance
$\exists x\,(\mathbf{T}(q,a,x) \wedge ax = bc)$	(A4) segment extension
$\mathbf{T}(a,b,a) \;\rightarrow\; a = b$	(A6) identity for betweenness

When using (A4) in OTTER, we Skolemize it:

$\mathbf{T}(q,a,ext(q,a,b,c)) \wedge E(a,ext(q,a,b,c),b,c)$ (A4) Skolemized

The original (1948) theory had the following additional fundamental properties of betweenness listed as axioms. (We follow the numbering of [12].)

$\mathbf{T}(a,b,b)$ (A12) Reflexivity for betweenness
$\mathbf{T}(a,b,c) \;\rightarrow\; \mathbf{T}(c,b,a)$ (A14) Symmetry for betweenness
$\mathbf{T}(a,b,d) \wedge \mathbf{T}(b,c,d) \;\rightarrow\; \mathbf{T}(a,b,c)$ (A15) Inner transitivity
$\mathbf{T}(a,b,c) \wedge \mathbf{T}(b,c,d) \wedge b \neq c \;\rightarrow\; \mathbf{T}(a,b,d)$ (A16) Outer transitivity
$\mathbf{T}(a,b,d) \wedge \mathbf{T}(a,c,d) \;\rightarrow\; \mathbf{T}(a,b,c) \vee \mathbf{T}(a,c,b)$ (A17) Inner connectivity
$\mathbf{T}(a,b,c) \wedge \mathbf{T}(a,b,d) \wedge a \neq b$ (A18) Outer connectivity
$\quad \rightarrow\; \mathbf{T}(a,c,d) \vee \mathbf{T}(a,d,c)$

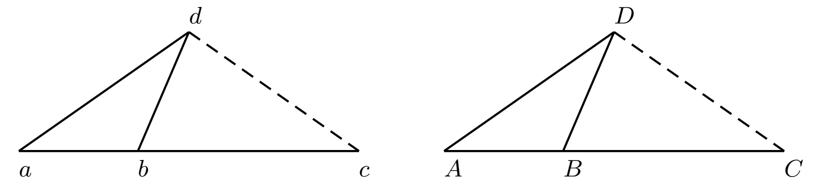
Of these only (A15) and (A18) appear in the 1959 version, because in 1956–57 Tarski and his students Kallin and Taylor showed that the other four are dependent (derivable from the remaining axioms). H. N. Gupta showed in his 1965 Ph. D. thesis [4] that (A18) is also dependent. The proof of (A18) is one of Quaife's challenge problems. Gupta also showed that (A15) implies (A6) using the other axioms of the 1959 system. Then one could have dropped (A6) as an axiom; but instead, Szmielew dropped (A15), keeping (A6) instead; then (A15) becomes a theorem.

All six of these axioms occur as theorems in [9]: (A12) is Satz 3.1, (A14) is Satz 3.2, (A15) is Satz 3.5, (A16) is Satz 3.7, (A18) is Satz 5.1, and (A17) is Satz 5.3. Hence, our research program of proving all the theorems in Szmielew's development using OTTER automatically captured these results as soon as we reached Satz 5.3.

2.4 The five-segment axiom

Hilbert [5] treated angles as primitive objects and angle congruence as a primitive relation, and took SAS (the side-angle-side triangle congruence principle) as an axiom. In Tarski's theory, angles are treated as ordered triples of points, and angle congruence is a defined notion, so a points-only formulation of the SAS principle is required. The key idea is Tarski's "five-segment axiom" (A5), shown in Fig. 1.

Fig. 1. The five-segment axiom (A5)

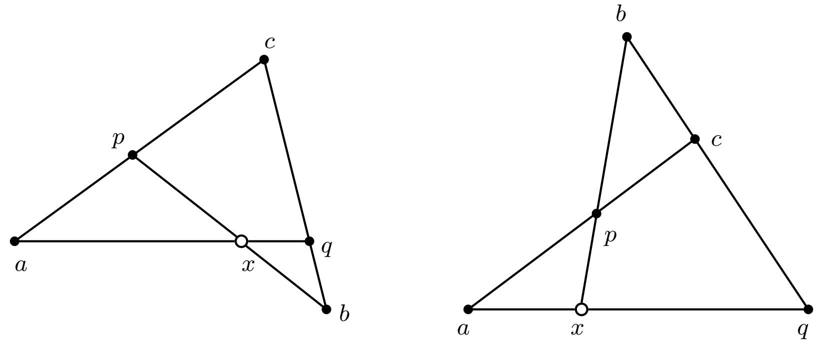


If the four solid segments in Fig. 1 are pairwise congruent, then the fifth (dotted) segments are congruent too. This is essentially SAS for triangles *dbc* and DBC. The triangles *abd* and ABD are surrogates, used to express the congruence of angles *dbc* and DBC. By using Axiom A5, we can avoid all mention of angles.

2.5 Pasch's axiom

Moritz Pasch [6] (See also [7], with an historical appendix by Max Dehn) supplied (in 1882) an axiom that repaired many of the defects that nineteenth-century rigor found in Euclid. Roughly, a line that enters a triangle must exit that triangle. As Pasch formulated it, it is not in $\forall\exists$ form. There are two $\forall\exists$ versions, illustrated in Fig. 2.5. These formulations of Pasch's axiom go back to Veblen [13], who proved outer Pasch implies inner Pasch. Tarski took outer Pasch as an axiom in [11].

Fig. 2. Inner Pasch (left) and Outer Pasch (right). Line *pb* meets triangle *acq* in one side. The open circles show the points asserted to exist on the other side.



$$\mathbf{T}(a,p,c) \wedge \mathbf{T}(b,q,c) \;\rightarrow\; \exists x\,(\mathbf{T}(p,x,b) \wedge \mathbf{T}(q,x,a)) \text{ (A7) inner Pasch}$$
$$\mathbf{T}(a,p,c) \wedge \mathbf{T}(q,c,b) \;\rightarrow\; \exists x\,(\mathbf{T}(a,x,q) \wedge \mathbf{T}(b,p,x)) \quad \text{outer Pasch}$$

For use in OTTER, we introduce Skolem symbols $ip(a,p,c,b,q)$ and $op(p,a,b,c,q)$ for the point x asserted to exist.

Tarski originally took outer Pasch as an axiom. In [4], Gupta proved both that inner Pasch implies outer Pasch, and that outer Pasch implies inner Pasch, using the other axioms of the 1959 system. The proof of outer Pasch from inner Pasch is one of Quaife's four challenge problems.

2.6 Dimension Axioms

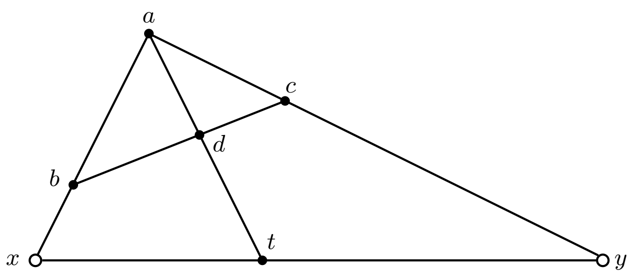
With no dimension axioms, Tarski's geometry axiomatizes theorems that are true in n-dimensional geometries for all n. For each positive integer n, we can specify that the dimension of space is at least n (with a lower dimension axiom A8$^{(n)}$), or at most n (with an upper dimension axiom A9$^{(n)}$). The upper dimension axiom says (in a first-order way) that the set of points equidistant from n given points is at most a line. The lower dimension axiom for n is the negation of the upper dimension axiom for $n-1$. For the exact statements of these axioms see [12].

2.7 Tarski's Parallel Axiom (A10)

In the diagram (Fig. 3), open circles indicate points asserted to exist.

$$\mathbf{T}(a,d,t) \wedge \mathbf{T}(b,d,c) \wedge a \neq d \;\rightarrow\; \exists x \exists y\,(\mathbf{T}(a,b,x) \wedge \mathbf{T}(a,c,y) \wedge \mathbf{T}(x,t,y) \text{ (A10)}$$

Fig. 3. Tarski's parallel axiom



The hypothesis says that t lies in the interior of angle a, as witnessed by b, c, and d. The conclusion says that some line through t meets both sides of the angle. Of course this fails in non-Euclidean geometry when both ab and ac are parallel to xy.

According to [12], Szmielew preferred to use the "triangle circumscription principle (A10_2) as the parallel axiom. Substituting (A10_2) was apparently one of the "inessential changes" made by Schwabhäuser. This principle says that if a, b, and c are not collinear, then there exists a point equidistant from all three.

2.8 Continuity axioms

Axiom schema (A11) is not a single axiom, but an axiom schema, essentially asserting that first-order Dedekind cuts are filled. Models of A1-A11 are all isomorphic to planes $\mathbb{F}^2$ where $\mathbb{F}$ is a real-closed field. One can also consider instead of (A11), the axioms of line-circle continuity and/or circle-circle continuity, which assert the existence of intersection points of lines and circles, or circles and circles, under appropriate hypotheses. None of the continuity axioms are used in the work reported on in this paper. Szmielew's development proceeds strictly on the basis of A1-A10.

3 Methodology

In this section we describe, with illustrative examples, the techniques we used in this project.

3.1 How OTTER works

Readers familiar with OTTER can skip this section. It is not an introduction to OTTER, but an attempt to make the subsequent information about our methodology comprehensible to those who do not have expertise with OTTER; at least, it should enable such readers to make sense of the input files and proofs we exhibit below and on the project's website [1]. For more information about OTTER, see [17].

OTTER is a clause-based resolution theorem prover. One writes `-A` for the negation of A. One writes `A | B` for disjunction ("or"). One does not write "and" at all, but instead one enters the two clauses separately. One writes $A \rightarrow B$ as `-A | B`. Similarly one writes $P \wedge Q \rightarrow R$ as `-P | -Q | R`.

Variables begin with `x,y,z,w,u,v`. Names beginning with any other letter are constants. A resolution theorem-prover requires the goal to be negated and entered as clauses. For example, to prove $A(x) \rightarrow \exists y\, B(x,y)$, we would enter the following clauses:

```
A(c).
-B(c,y).
```

After proving this theorem, if we want to use it to prove the next theorem, we invent a new Skolem symbol `f` and enter the theorem as `-A(x) | B(x,f(x))`.

The input to OTTER is contained in two lists, the "set of support" (sos) and "usable". The fundamental run-time loop of OTTER moves a clause from sos to usable, and then tries to use one of the specified inference rules to generate new clauses from that clause and other clauses in usable. If conclusions are generated, OTTER has to decide whether to keep them or not. If it decides to keep them, they are placed on sos, where they can eventually be used to generate yet more new clauses. If the empty clause is generated, that means a proof has been found, and it will be output.

The fundamental problem of automated deduction is to avoid drowning in a sea of useless conclusions before finding the desired proof. One tries to get control over this by assigning "weights" to clauses, adjusting those weights in various ways, and using them to control both which clauses are kept, and which clause is selected from sos for the next iteration of the loop. By default: the weight of a clause is the number of its symbols; the next clause selected is the lightest one in sos; and clauses are kept if their weight does not exceed a parameter `max_weight`. More sophisticated ways of setting the weights have been developed over the past decades and are discussed below. The idea is to get the weights of the important clauses to be small, and then to squeeze down `max_weight` to prevent drowning.

In addition to techniques involving weighting, there are other ways to control OTTER's search:

- Use a propitious combination of rules of inference. For an introduction to these rules please refer to [17].
- You have some control over which clause will be selected from sos at the next iteration, using OTTER's `pick_given_ratio`.
- You have some control over how the search starts and what kind of proof you want to look for (forward, backward, or bi-directional) by choosing which of your clauses to put in sos and which to put in usable.

3.2 Hints

Putting a clause into `list(hints)` causes OTTER to give that clause, if deduced, a low weight, causing it to be retained, even if its default weight would have been so large as to cause it to be discarded. One has options (specified at the top of an OTTER file) to cause this weight adjustment to apply to clauses that match the hints, or subsume the hints, or are subsumed by the hints. The way we use hints is describe in Section 3.5. The technique of hints was invented by Veroff [14] and later incorporated into OTTER. As a technical note: when using hints, you should always include these lines, without which your hints will not have the desired effect.

```
assign(bsub_hint_wt,-1).
set(keep_hint_subsumers).
```

Another similar technique is known as *resonators.* This is more useful when one has a proof in hand, and wishes to find a shorter proof. For the exact differences between hints and resonators, see [16], p. 259.

3.3 Giving OTTER the diagram

This refers to a technique we learned from Quaife [8]. Suppose we are trying to find an x such that $A(a, b, x)$. If we let OTTER search, without making any attempt to guide the search, essentially OTTER will generate all the points that can be constructed from a and b (and other points in the problem or previously constructed in clauses that were kept) by the Skolem functions for segment extension and inner Pasch. Imagine trying to prove a geometry theorem that way: just take your ruler and draw all the lines you can between the points of the diagram, and label all the new points formed by their intersections (that is using Pasch), and construct every point that can be constructed by extending a segment you have by another segment you have, and see if any of the new points is the point you are trying to construct, and if not, repeat. You will generate a sea of useless points, even if you discard those with too many construction steps.

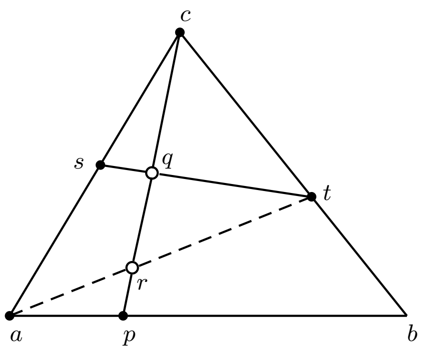
In order to guide OTTER to construct the right points, we "give it the diagram" by defining the points to be constructed, using the Skolem functions. For example, consider the "crossbar theorem" (Satz 3.17). See Fig. 4, which shows the diagram and the Otter input (not shown are the axioms A1-A6 and the previous theorems, which are placed in `list(usable)`). The two lines defining r and q are what we call "giving OTTER the diagram." With those lines present, OTTER finds a proof instantly. Remove them, and OTTER does not find a proof (at least, not in ten minutes).

The reason this works is that q, being a single letter, has weight 1, so clauses involving q have much lower weight than would the same clauses with q replaced by $ip(c, s, a, t, r)$, which has weight 6. A clause involving the terms defining both q and r would have weight at least 14, so it would not be considered very soon, and meantime, many other points will be constructed.

In this simple example, if you put all clauses into sos, and nothing in usable, OTTER does eventually find a proof without being given the diagram, but it is certainly much slower than with the diagram. In other, more complex examples, we think this technique is essential. Here, for example, are the lines from the

Fig. 4. The crossbar theorem asserts that q exists, given the other points except r. To prove it we construct first r and then q, using inner Pasch twice.

```
list(sos).
T(a,s,c).
T(b,t,c).
T(a,p,b).
-T(p,x,c)|-T(s,x,t).
r = ip(c,t,b,a,p).
q = ip(c,s,a,t,r).
end_of_list.
```



input file for Satz 5.1, the inner connectivity of betweenness, describing the two rather complicated diagrams on pp. 39–40 of [9]:

```
c1=ext(a,d,c,d).
d1=ext(a,c,c,d).
p=ext(c1,c,c,d).
r=ext(d1,c,c,e).
q=ext(p,r,r,p).
b1 = ext(a,c1,c,b).
b2 = ext(a,d1,d,b).
e = ip(c1,d,b,d1,c).
```

3.4 Assistance with proof by cases

We found that OTTER often “got stuck” when Szmielew’s proof required an argument by cases. We could sometimes get around this difficulty by simply adding a clause `A | -A`, where the proof needs to proceed by cases on A. It seems that OTTER prefers constructive proofs! This technique is called “tautology adjunction” by the second author, who used it decades ago in proving that subgroups of index 2 are normal. We used this in many input files. Here we discuss just one example. The inner connectivity of betweenness (A17 above, Satz 5.3 in Szmielew) is derived as an easy corollary of Satz 5.1, which is

$$a \neq b \wedge \mathbf{T}(a,b,c) \wedge \mathbf{T}(a,b,d) \ \rightarrow\ \mathbf{T}(a,c,d) \vee \mathbf{T}(a,d,c).$$

The natural way to formulate `list(sos)` for this problem would be

```
a != b.
T(a,b,c).
T(a,b,d).
-T(a,c,d).
-T(a,d,c).
```

Of course, that does not suffice to find a proof. So, we added the description of the diagram, as given in the previous section. Unfortunately OTTER could still not find a proof, even with hints.

The proof in Szmielew proceeds by cases. The methodology we followed in such cases was this:

- Add one case to sos, e.g. `c=c1`. (`c1` is a constant from the diagram, above.)
- If we find a proof: add the steps of that proof as hints.
- Remove that case, and add its negation, e.g. `c != c1`
- If we find a proof: add its steps also as hints.
- Now remove both cases and add their disjunction: `c = c1 | c != c1`.

Often we could find a proof. The amazing thing, to us, was that the same file with the tautology deleted would often *not* find a proof. We do not understand exactly why tautology adjunction works, but in practice, it often helps.

The example at hand required two divisions into cases (so tautology disjunction was applied recursively). The first is the case whether `c=c1` or not, and the second, whether `d1=e` or not. In the input file, one can see in the two commented lines in `list(sos)` the traces of completing the last argument by cases.

```
% d1 = e.
% d1!= e.
d1=e | d1!=e.
c = c1 | c != c1.
```

We do not mean to imply that this was all there was to proving Satz 5.1. This was just the last difficulty. By that time, the input file already contained a long list of hints obtained by methods described in the next section. The final proof has 131 steps.

3.5 Supplying proof steps

Our methodology was as follows: when trying to prove a theorem, we prepared an input file with the negated goal in `list(sos)`, and the axioms and previously proved theorems in list(usable), and the choice of inference rules as described below. If this did not find a proof, and the proof in Szmielew had a diagram, we supplied the diagram. If that still did not work, then we tried supplying some intermediate goals. We would list some important steps of the proof from Szmielew (in negated form) in `list(passive)`, with answer literals having numbers for those goals. When this file is run, one sees which of the intermediate goals are being proved. Among other things, this showed us where OTTER was "getting stuck." But even though we found no proof of the main goal, we had the proofs of some intermediate goals. We converted these proof steps to hints, and ran OTTER again. Sometimes more intermediate goals would be reached. One can tinker with `max_weight`: sometimes a smaller `max_weight` may keep one from drowning, and sometimes a larger `max_weight` may allow a vital clause to be kept. With luck, this process would converge.

We note a technical point: Exactly how does one extract a list of hints from the proofs in an OTTER output file? The lines of a proof contain extraneous material, such as line numbers of the ancestors and justifications of the inferencea, that must be stripped out. Since this has to be done iteratively and often, and sometimes the list of clauses to be converted is long, one needs an automated method to do this. The first author used a PHP script, and the second author used a Unix shell script, for this purpose.

3.6 Divide and conquer

If the methods described above still did not produce a proof, we tried to cut the task in half as follows: we picked one of the intermediate steps (already in `list(passive)`, but not being proved), and added its positive form to `list(sos)`. In the abstract, say we are trying to prove conclusion C from hypothesis A. The proof, we think, should proceed by first proving B from A, and then proving C from A and B. If we can't succeed in proving C from A in one run, it makes sense to assume B as well as A, and try to prove C. In a separate run, we try to prove B from A. That may be easier with the negated form of B in `list(sos)` instead of `list(passive)`.

3.7 Choice of inference rules and settings

We mentioned above that one of the ways OTTER can be controlled is through a propitious choice of inference rules and settings. We tinkered with our choices often, in the hope that a different choice would be better. We did not find that one choice was always best. We always used hyperresolution, and we sometimes used unit resolution or binary resolution. Sometimes it helped to turn binary resolution on. Sometimes it helped to turn binary resolution off. Generally, we think it worked better with both hyperresolution and binary resolution allowed, but not always. Occasionally we could "break through" using `sos_queue` or `set(input_sos_first)`. We *often* changed the values of the parameters `max_weight`, `max_distinct_vars`, `pick_given_ratio`, and `max_proofs`. We used paramodulation for equality reasoning. The second author believes that failing to use paramodulation was an important reason for the limited success of the experiments he made in the 1980s in Tarskian geometry; but we also note that Quaife writes that paramodulation was available for use but seldom actually used in his proofs.

3.8 What about Prover9? Or E, Spass, Vampire?

The question inevitably arises, why did we use OTTER instead of the newer provers mentioned? The short answer is, we did use Prover9, and others tried E, Spass, and Vampire. Whenever we could not get a proof with OTTER, the first author would try it in Prover9. But in every case, Prover9 also could not find a proof; and sometimes Prover9 would not succeed when OTTER would. In so-far unpublished work, Narboux *et. al.* tried E, Spass, and Vampire on the theorems from Szmielew, with a 14% success rate. This was in a completely automated mode, so not directly comparable to the 100% success rate we achieved with a human-OTTER team.

4 Results

All the input files and resulting OTTER proofs that we found are posted on the web at [1]. Here we list some of the more difficult proofs we found, with the length of the proofs. It is not meaningful to list the execution times, as the posted input files (for the difficult proofs) were developed iteratively, and contain many hints based on previous runs, so most of the posted input files produce a proof more or less immediately.

4.1 Properties of betweenness

In section 2.3, we listed six difficult theorems (A12-A18), each of which Tarski originally took as an axiom, and their numbers as theorems in [9]. There is another theorem about betweenness that occurs as a challenge problem in [15], namely the "five-point theorem":

$$\mathbf{T}(z,w,v) \wedge \mathbf{T}(z,y,v) \wedge \mathbf{T}(w,x,y) \;\rightarrow\; \mathbf{T}(z,x,v).$$

We found OTTER proofs of all those theorems. The following table gives the length of these proofs, which is perhaps some indication of the relative difficulty of finding them.

A12	Satz 3.1	Reflexivity for betweenness	4 steps
A14	Satz 3.2	Symmetry for betweenness	4 steps
A15	Satz 3.5	Inner transitivity	4 steps
A16	Satz 3.7	Outer transitivity	16 steps
A17	Satz 5.3	Inner connectivity	131 steps
A18	Satz 5.1	Outer connectivity	4 steps

4.2 Midpoints, perpendiculars, and isosceles triangles

The "midpoint theorem" asserts that every segment has a midpoint. The traditional Euclidean construction involves the intersection points of two circles, but we are required to prove the theorem from A1-A9. (Not even the parallel axiom A10 is to be used.) This is a difficult problem, and was apparently not solved until Gupta's 1965 thesis [4]. Two important preliminary steps are the erection of a perpendicular to a line at a given point, and the "Lotsatz", which says we can drop a perpendicular to a line from a point not on the line. Remember this must be done without circles! A very clever observation of Gupta was that it is easy to construct the midpoint of ab if ab is the base of an isosceles triangle (only two applications of inner Pasch are needed). This plays a key role in the proof of the Lotsatz. The two theorems on perpendiculars are used to construct the midpoint. Of course, once we have midpoints and perpendiculars, it is trivial to show that every segment is the base of an isosceles triangle; that theorem does not even occur explicitly in [9]. An important lemma used in the proofs of these theorems is the “Krippenlemma” (too long to state here). Here are the lengths of our proofs of these theorems:

Satz 7.22	Krippenlemma	132 steps
Satz 7.25	Base of isosceles triangle has a midpoint	123 steps
Satz 8.18	Lotsatz: there is a perpendicular to a line from a point not on the line	332 steps
Satz 8.21a	There is a perpendicular to a line through a point on the line on the opposite side from a given point not on the line.	108 steps
Satz 8.22b	Given segment ab and perpendiculars ap and qb, and point t on line ab between p and q, with $ap \leq qb$, then segment ab has a midpoint.	233 steps
Satz 8.22	Every segment has a midpoint	23 steps

4.3 The diagonals of a rhombus bisect each other

One of the challenges in [15], solved by Quaife, was to prove that the diagonals of a rectangle bisect each other. That problem assumed that opposite sides are equal and the diagonals meet in the midpoint of one diagonal; you are to prove that the other diagonal is also bisected. A more general problem is found in Satz 7.21, which asserts that in a quadrilateral with opposite sides equal, the diagonals meet, and bisect each other. Our proof of this theorem has 31 steps.

4.4 Inner and outer Pasch

The proof that inner Pasch implies outer Pasch (using A1-A6 and A8) was one of the major results of Gupta’s thesis, and enabled Szmielew to replace outer Pasch by inner Pasch as an axiom. This theorem was one of Quaife’s four challenges. It is Satz 9.6 in [9]. The proof posted on our archive is 111 steps, preceded by proofs Satz 9.4 and Satz 9.5 of 57 and 45 steps. Satz 9.5 is the “plane separation theorem”, important in its own right.

4.5 Hilbert’s axioms

Hilbert’s theory can be interpreted in Tarski’s, using pairs of points for lines and ordered triples of points for angles and planes. His axioms (so interpreted) all turn out to be either axioms, theorems proved in [9], or extremely elementary consequences of theorems proved in [9]. The theorems of [9] needed are 2.3,2.4,2.5,2.8; 3.2,3.13;6.16, 6.18; 8.21, 8.22; 9.8, 9.25, 9.26, 11.15, 11.49, and finally Hilbert’s parallel axiom is Satz 12.11. We have posted OTTER proofs of all these theorems. Narboux and Braun have proof-checked Hilbert’s axioms in Tarskian geometry, using Coq [2].

5 Discussion

5.1 Proof checking *vs.* proof finding

"Proof checking" refers to obtaining computer-verified proofs, starting with human-written proofs. "Proof finding" refers to the traditional task of automated deduction, finding a proof by searching a large space of possible proofs, either without possessing a proof or without making use of a known proof. The use of hints blurs this distinction, as we shall now explain. If we have a proof in hand (whether generated by human or machine), and we enter its steps as hints, with a low `max_weight`, we force OTTER to find a proof containing mostly the same formulas as the proof in the hints. (The order of deductions might be different.) Thus we can almost always ensure that Otter finds a proof, if we have a proof in hand. One could plausibly claim that this is proof-checking, not proof-finding.

What if, instead of entering all the steps as hints, we enter some key steps in `list(passive)`, and generate some proofs of some of those steps, and put those steps in as hints? Now are we doing proof-checking, or proof-finding? What may have appeared to be a clearcut distinction turns out to be a continuum of possibilities.

5.2 1992 *vs.* 2014

The research reported here shows how much progress has occurred in automated reasoning in that time period. Indeed, approximately thirty years ago, almost all of the theorems cited in this article were out of reach. The question arises, whether this advance might be due simply to the increased memory capacity and speed of modern computers. Perhaps Quaife, equipped with one of our computers, would have found these proofs? Perhaps we, constrained to run on a computer from 1990, might not have found them? We argue that this is not the case: the improvements are due not to faster hardware but to the techniques described above, namely generating partial proofs (of intermediate steps) and using their steps as hints; using the right combination of inference rules and settings; using tautology adjunction to help with proofs by cases; and divide-and-conquer. We note that Quaife did have (in fact, invented) the technique of giving OTTER the diagram. We did not actually try to run on a 1990 computer, and we do not doubt that it would have been painful and discouraging; but we think the main credit should go to Veroff's invention of hints, and the uses of hints developed by the second author and applied here.

6 Conclusions

We used OTTER to find proofs of the theorems in Tarskian geometry in the first nine chapters of Szmielew's development in Part I of [9]. Those theorems include the four unsolved challenge problems from Quaife's book[8], and the verification of Hilbert's axioms.

Input files and the resulting proofs for all the theorems that we have proved from Szmielew are archived at [1], where they can be viewed or downloaded. The second author has also conducted many experiments aimed at shortening some of these proofs or finding forward proofs instead of backwards or bidirectional proofs; we have not reported on those experiments here.

References

1. Beeson, M.: The Tarski formalization project, http://www.michaelbeeson.com/research/FormalTarski/index.php
2. Braun, G., Narboux, J.: From Tarski to Hilbert. In: Ida, T., Fleuriot, J. (eds.) Automated Deduction in Geometry 2012 (2012)
3. Caviness, B.F., Johnson, J.R. (eds.): Quantifier Elimination and Cylindrical Algebraic Decomposition. Springer, Wien/New York (1998)
4. Gupta, H.N.: Contributions to the Axiomatic Foundations of Geometry. Ph.D. thesis, University of California, Berkeley (1965)
5. Hilbert, D.: Foundations of Geometry (Grundlagen der Geometrie). Open Court, La Salle, Illinois (1960), second English edition, translated from the tenth German edition by Leo Unger. Original publication date, 1899.
6. Pasch, M.: Vorlesung über Neuere Geometrie. Teubner, Leipzig (1882)
7. Pasch, M., Dehn, M.: Vorlesung über Neuere Geometrie. B. G. Teubner, Leipzig (1926), the first edition (1882), which is the one digitized by Google Scholar, does not contain the appendix by Dehn.
8. Quaife, A.: Automated Development of Fundamental Mathematical Theories. Springer, Berlin Heidelberg New York (1992)
9. Schwabhäuser, W., Szmielew, W., Tarski, A.: Metamathematische Methoden in der Geometrie: Teil I: Ein axiomatischer Aufbau der euklidischen Geometrie. Teil II: Metamathematische Betrachtungen (Hochschultext). Springer–Verlag (1983), reprinted 2012 by Ishi Press, with a new foreword by Michael Beeson.
10. Tarski, A.: A decision method for elementary algebra and geometry. Tech. Rep. R-109, second revised edition, reprinted in [3], pp. 24–84, Rand Corporation (1951)
11. Tarski, A.: What is elementary geometry? In: Henkin, L., Suppes, P., Tarksi, A. (eds.) The axiomatic method, with special reference to geometry and physics. Proceedings of an International Symposium held at the Univ. of Calif., Berkeley, Dec. 26, 1957–Jan. 4, 1958. pp. 16–29. Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam (1959), available as a 2007 reprint, Brouwer Press, ISBN 1-443-72812-8
12. Tarski, A., Givant, S.: Tarski's system of geometry. The Bulletin of Symbolic Logic 5(2), 175–214 (June 1999)
13. Veblen, O.: A system of axioms for geometry. Transactions of the American Mathematical Society 5, 343–384 (1904)
14. Veroff, R.: Using hints to increase the effectiveness of an automated reasoning program. Journal of Automated Reasoning 16(3), 223–239 (1996)
15. Wos, L.: Automated reasoning: 33 basic research problems. Prentice Hall, Englewood Cliffs, New Jersey (1988)
16. Wos, L.: Automated reasoning and the discovery of missing and elegant proofs. Rinton Press, Paramus, New Jersey (2003)
17. Wos, L., Pieper, G.W.: A fascinating country in the world of computing. World Scientific (1999)