Journal of Automated Reasoning **5** (1989), 239–255.

Problem Corner:

Automated Reasoning about Elementary Point-Set Topology★

CYNTHIA A. WICK and WILLIAM W. McCUNE
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439-4801, U.S.A.

(Received: 6 December 1988)

Abstract. In this paper we present first-order formulas for basic point-set topology, in an attempt to extend the mathematical range available for exploration with automated theorem-proving programs. We present topology definitions and sample lemmas both in first-order logic and in clausal form. We then illustrate some of the difficulties of these sample lemmas through a proof of a basic lemma in five parts.

Key words. Automated reasoning, automated theorem proving, set theory, topology.

1. Introduction

This note is intended to be in the same spirit as the set theory paper by Boyer *et al.* [2]. That paper presents a finite first-order formulation of full set theory based on Gödel's axiomatization [3]. The aims of that paper were to provide the basis for an unlimited supply of truly difficult problems and to challenge the automated deduction community to experiment in that direction.

The focus of this paper is elementary point-set topology, a field built upon part of set theory, and a field that has many of the obstacles to automation discussed in the set theory paper. But, because of strict typing and the limited use of the set theory (see Section 4), we believe that the test problems presented here may be easier to handle than those presented in the set theory paper.

Point-set topology arose as a generalization of the study of continuous functions on the real line. In this paper we look at basic concepts such as open and closed sets, subspaces, bases, and limit points, but we do not get as far as continuous functions or metrics. There is no inherent difficulty in considering the more advanced concepts, but we believe that the basic concepts provide sufficient challenge to current and proposed automated deduction systems. We closely follow the development given by Munkres [7], which is similar to Kelley's development [4].

★ This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38, and by the Division of Educational Programs, Argonne National Laboratory.

The paper is organized in the following way. Section 2 contains definitions of basic topology concepts as first-order formulas in set theory notation. Axioms and definitions of set theory concepts such as element, subset, intersection, and relative complement are omitted. In Section 3 we give a set of lemmas, in the same notation as the Section 2 definitions, which should be formally provable from the Section 2 definitions and the appropriate set theory axioms and definitions. Section 4 contains the clause form of the Section 2 definitions in strict clause notation (prefix functions and predicates). We have renamed some of the functions and predicates in order to take advantage of object types – for example, in the clauses we use el_p(x, xa) when x is a point and xa is a set, and we use el_s(xa, xt) when xa is a set and xt is a collection (a set of sets of points). Section 5 contains the clause form of the negations of the Section 3 lemmas. In Section 6 we give a five-part resolution proof of the first lemma.

2. Definitions

We use a sorted first-order logic, and for the most basic topology concepts, we have three disjoint types of object: points, sets of points (called sets), and sets of sets of points (called collections). The only concept we consider (in this paper) that does not fit into one of those sorts is function.

A departure from the set theory paper [3] is that we do not worry about sets being 'small' enough (the $m(x)$ conditions in [3]) to be members of other sets. We believe that our restricted use of set theory – in particular, the hierarchy of types point, set, collection – protects us from paradoxes, at least for this beginning study of topology.

We include definitions of compactness and of continuous function, but they are not mentioned elsewhere in the paper. They are included for the curious because they, along with connectedness, are essential for any meaningful study of topology. Compactness requires the concept of finiteness, and continuous function requires the concepts of function and of preimage of a set, but those three additional concepts are not included because we wish to keep the focus on point-set topology.

We omit definitions of most set theory concepts. The following is a list of symbols that occur in the definitions and in the lemmas in Section 4, but are neither axiomized nor defined here.

$=, \in, \subseteq, \emptyset, \cup, \cap$, *disjoint*, – (relative complement), *finite*, *function*, *preimage*.

In the following definitions, we use the rule that symbols starting with [u, v, w, x, y, z, ...] represent points, [A, B, C, U, V, W, X, Y, Z, ...] represent sets, and [F, G, H, S, T, ...] represent collections. All variables are quantified.

Sigma (union of members)

$$\forall F\, \forall u\left(u \in sigma(F) \leftrightarrow \exists A \begin{bmatrix} u \in A\ \wedge \\ A \in F \end{bmatrix}\right)$$

Pi (intersection of members)

$$\forall F\, \forall u(u \in pi(F) \leftrightarrow \forall A(A \in F \rightarrow u \in A))$$

Topological space

$$\forall X\, \forall T \left(top\_space(X, T) \leftrightarrow \begin{bmatrix} sigma(T) \subseteq X\ \wedge \\ \emptyset \in T\ \wedge \\ X \in T\ \wedge \\ \forall Y\, \forall Z(Y \in T \wedge Z \in T \rightarrow (Y \cap Z) \in T)\ \wedge \\ \forall F(F \subseteq T \rightarrow sigma(F) \in T) \end{bmatrix} \right)$$

Open set

$$\forall U\, \forall X\, \forall T \left(open(U, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T)\ \wedge \\ U \in T \end{bmatrix} \right)$$

Closed set

$$\forall U\, \forall X\, \forall T \left(closed(U, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T)\ \wedge \\ open(X - U, X, T) \end{bmatrix} \right)$$

Finer topology

$$\forall T\, \forall S\, \forall X \left(finer(T, S, X) \leftrightarrow \begin{bmatrix} top\_space(X, T)\ \wedge \\ top\_space(X, S)\ \wedge \\ S \subseteq T \end{bmatrix} \right)$$

Basis for a topology

$$\forall X\, \forall F \left(basis(X, F) \leftrightarrow \begin{bmatrix} sigma(F) = X\ \wedge \\ \forall y\, \forall B_1\, \forall B_2 \left(\begin{bmatrix} y \in X\ \wedge \\ B_1 \in F\ \wedge \\ B_2 \in F\ \wedge \\ y \in (B_1 \cap B_2) \end{bmatrix} \rightarrow \exists B_3 \begin{bmatrix} y \in B_3\ \wedge \\ B_3 \in F\ \wedge \\ B_3 \subseteq (B_1 \cap B_2) \end{bmatrix} \right) \end{bmatrix} \right)$$

Topology generated by a basis

$$\forall F\, \forall U \left(U \in top\_of\_basis(F) \leftrightarrow \forall x \left(x \in U \rightarrow \exists Z \begin{bmatrix} x \in Z\ \wedge \\ Z \in F\ \wedge \\ Z \subseteq U \end{bmatrix} \right) \right)$$

Subspace topology

$$\forall X\, \forall T\, \forall Y\, \forall U \left(U \in subspace\_top(X, T, Y) \leftrightarrow \begin{bmatrix} top\_space(X, T)\ \wedge \\ Y \subseteq X\ \wedge \\ \exists V(V \in T \wedge U = (Y \cap V)) \end{bmatrix} \right)$$

Interior of a set

$$\forall Y\,\forall X\,\forall T\,\forall u\left(u\in interior(Y,X,T)\leftrightarrow\begin{bmatrix}top\_space(X,T)\ \wedge\\ Y\subseteq X\ \wedge\\ \exists V(u\in V\wedge V\subseteq Y\wedge open(V,X,T))\end{bmatrix}\right)$$

Closure of a set

$$\forall Y\,\forall X\,\forall T\,\forall u\left(u\in closure(Y,X,T)\leftrightarrow\begin{bmatrix}top\_space(X,T)\ \wedge\\ Y\subseteq X\ \wedge\\ \forall V(Y\subseteq V\wedge closed(V,X,T)\rightarrow u\in V)\end{bmatrix}\right)$$

Neighborhood (Note that a neighborhood of y is defined here as an open set containing y [7]. Another definition is that a neighborhood of y is any superset of an open set containing y [4].)

$$\forall U\,\forall y\,\forall X\,\forall T\left(neighborhood(U,y,X,T)\leftrightarrow\begin{bmatrix}top\_space(X,T)\ \wedge\\ open(U,X,T)\ \wedge\\ y\in U\end{bmatrix}\right)$$

Limit point

$$\forall z\,\forall Y\,\forall X\,\forall T\Bigg(limit\_pt(z,Y,X,T)\leftrightarrow$$

$$\begin{bmatrix}top\_space(X,T)\ \wedge\\ Y\subseteq X\ \wedge\\ \forall U(neighborhood(U,z,X,T)\rightarrow\exists w(w\in(U\cap Y)\wedge w\neq z))\end{bmatrix}\Bigg)$$

Boundary of a set

$$\forall Y\,\forall X\,\forall T\,\forall u\left(u\in boundary(Y,X,T)\leftrightarrow\begin{bmatrix}top\_space(X,T)\ \wedge\\ u\in closure(Y,X,T)\ \wedge\\ u\in closure(X-Y,X,T)\end{bmatrix}\right)$$

Hausdorff space

$$\forall X\,\forall T\Bigg(hausdorff(X,T)\leftrightarrow\Bigg[\begin{matrix}top\_space(X,T)\ \wedge\\ \forall x_1\,\forall x_2\end{matrix}\quad\left(\begin{bmatrix}x_1\in X\ \wedge\\ x_2\in X\ \wedge\\ x_1\neq x_2\end{bmatrix}\rightarrow\right.$$

$$\left.\exists U_1\,\exists U_2\begin{bmatrix}neighborhood(U_1,x_1,X,T)\ \wedge\\ neighborhood(U_2,x_2,X,T)\ \wedge\\ disjoint(U_1,U_2)\end{bmatrix}\right)\Bigg]\Bigg)$$

Separation in a topological space

$$\forall A_1 \forall A_2 \forall X \forall T \left(separation(A_1, A_2, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ A_1 \neq \emptyset \wedge A_2 \neq \emptyset \wedge \\ A_1 \in T \wedge A_2 \in T \wedge \\ (A_1 \cup A_2) = X \wedge disjoint(A_1, A_2) \end{bmatrix} \right)$$

Connected topological space

$$\forall X \forall T \left(connected(X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ \neg \exists A_1 \exists A_2 \, separation(A_1, A_2, X, T) \end{bmatrix} \right)$$

Connected set

$$\forall A \forall X \forall T \left(connected\_set(A, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \wedge \\ connected(A, subspace\_top(X, T, A)) \end{bmatrix} \right)$$

Open covering

$$\forall F \forall X \forall T \left(open\_cover(F, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ F \subseteq T \wedge \\ sigma(F) = X \end{bmatrix} \right)$$

Compact topological space

$$\forall X \forall T \left(compact(X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ \forall F_1 \left(open\_cover(F_1, X, T) \rightarrow \exists F_2 \begin{bmatrix} finite(F_2) \wedge \\ F_2 \subseteq F_1 \wedge \\ open\_cover(F_2, X, T) \end{bmatrix} \right) \end{bmatrix} \right)$$

Compact set

$$\forall A \forall X \forall T \left(compact\_set(A, X, T) \leftrightarrow \begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \wedge \\ compact(A, subspace\_top(X, T, A)) \end{bmatrix} \right)$$

Continuous function

$$\begin{matrix} \forall F \forall X_1 \forall T_1 \\ \forall X_2 \forall T_2 \end{matrix} \left(cont\_func(F, X_1, T_1, X_2, T_2) \leftrightarrow \begin{bmatrix} top\_space(X_1, T_1) \wedge \\ top\_space(X_2, T_2) \wedge \\ function(F, X_1, X_2) \wedge \\ \forall V (open(V, X_2, T_2) \rightarrow \quad open(preimage(F, V), X_1, T_1)) \end{bmatrix} \right)$$

3. Lemmas

The following lemmas should be formally provable from the definitions in the preceding section and the appropriate set theory axioms and definitions. A five-part resolution proof of Lemma 1 appears in Section 6.

1. The topology generated by a basis gives rise to a topological space.

$$\forall F\, \forall X(basis(X, F) \rightarrow top\_space(X, top\_of\_basis(F)))$$

2. If (X, T) is a topological space, A is a subset of X, and every point in A has a neighborhood U that is a subset of A, then A is open in (X, T).

$$\forall X\, \forall T\, \forall A\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \wedge \\ \forall y\left(y \in A \rightarrow \exists U\begin{bmatrix} neighborhood(U, y, X, T) \wedge \\ U \subseteq A \end{bmatrix}\right) \end{bmatrix} \rightarrow open(A, X, T)\right)$$

3. The subspace topology gives rise to a topological space.

$$\forall X\, \forall T\, \forall Y(top\_space(X, T) \wedge Y \subseteq X \rightarrow top\_space(Y, subspace\_top(X, T, Y)))$$

4. If Y is open in X, and A is open in Y, then A is open in X.

$$\forall X\, \forall T\, \forall Y\, \forall A(open(Y, X, T) \wedge open(A, Y, subspace\_top(X, T, Y)) \rightarrow open(A, X, T))$$

5. A finer topology induces a finer subspace topology.

$$\forall T_1\, \forall T_2\, \forall X\, \forall Y(finer(T_1, T_2, X) \wedge Y \subseteq X \rightarrow finer(subspace\_top(X, T_1, Y), subspace\_top(X, T_2, Y), X))$$

6. An alternative definition of top_of_basis.

$$\forall F\, \forall U\left(U \in top\_of\_basis(F) \leftrightarrow \exists G\begin{bmatrix} G \subseteq F \wedge \\ U = sigma(G) \end{bmatrix}\right)$$

7. Arbitrary intersections and finite unions of closed sets are closed.

$$\forall X\, \forall T\left(top\_space(X, T) \rightarrow \begin{bmatrix} closed(\emptyset, X, T) \wedge \\ closed(X, X, T) \wedge \\ \forall Y_1\, \forall Y_2\left(\begin{bmatrix} closed(Y_1, X, T) \wedge \\ closed(Y_2, X, T) \end{bmatrix} \rightarrow closed(Y_1 \cup Y_2, X, T)\right) \wedge \\ \forall F\left(\begin{bmatrix} sigma(F) \subseteq X \wedge \\ \forall V(V \in F \rightarrow closed(V, X, T)) \end{bmatrix} \rightarrow closed(pi(F), X, T)\right) \end{bmatrix}\right)$$

8. The interior of A is a subset of A, which is a subset of the closure of A.

$$\forall A\ \forall X\ \forall T\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \end{bmatrix} \rightarrow \begin{bmatrix} interior(A, X, T) \subseteq A \wedge \\ A \subseteq closure(A, X, T) \end{bmatrix}\right)$$

9. If A is open, the interior of A is A, and if A is closed, the closure of A is A.

$$\forall A\ \forall X\ \forall T\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \end{bmatrix} \rightarrow \begin{bmatrix} open(A, X, T) \leftrightarrow A = interior(A, X, T) \wedge \\ closed(A, X, T) \leftrightarrow A = closure(A, X, T) \end{bmatrix}\right)$$

10. The interior and the boundary of a set are disjoint.

$$\forall A\ \forall X\ \forall T\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \end{bmatrix} \rightarrow interior(A, X, T) \cap boundary(A, X, T) = \emptyset\right)$$

11. The union of the interior and the boundary is the closure.

$$\forall A\ \forall X\ \forall T\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \end{bmatrix} \rightarrow interior(A, X, T) \cup boundary(A, X, T) = closure(A, X, T)\right)$$

12. If the boundary of A is empty, A is both open and closed.

$$\forall A\ \forall X\ \forall T\left(\begin{bmatrix} top\_space(X, T) \wedge \\ A \subseteq X \end{bmatrix} \rightarrow \left(boundary(A, X, T) = \emptyset \leftrightarrow \begin{bmatrix} open(A, X, T) \wedge \\ closed(A, X, T) \end{bmatrix}\right)\right)$$

13. If some limit points are added to a connected set, the result is connected.

$$\forall A\ \forall B\ \forall X\ \forall T\left(\begin{bmatrix} connected\_set(A, X, T) \wedge \\ \forall y(y \in B \rightarrow limit\_pt(y, A, X, T)) \end{bmatrix} \rightarrow connected\_set(A \cup B, X, T)\right)$$

14. The closure of a connected set is connected. (Note that Lemma 13 ⇒ Lemma 14.)

$$\forall A\ \forall X\ \forall T(connected\_set(A, X, T) \rightarrow connected\_set(closure(A, X, T), X, T))$$

4. Definitions in Clause Form

When we started to experiment with some of the test problems, we encountered the difficulty of typed objects, because none of our theorem provers handles typed variables. If one ignores variable types, atoms such as element(sigma(F), A), in which sigma(F) and A are both sets, can be derived by resolution. Such atoms not only pollute the search space, but also raise the question of soundness – what would it mean for such an atom to occur in a refutation? We considered several options for solving these problems.

1. Include type literals for the types point, set, and collection. In the Section 2 definitions and the Section 3 lemmas, for each universally quantified variable of type (for example) point one would replace $\forall xF$ with $\forall x(point(x) \rightarrow F)$, and for each existentially quantified variable of type (for example) set one would replace $\exists XF$ with $\exists X(set(X) \wedge F)$.
2. Use the trick of type functions around terms to prevent unification of terms of different types. Each constant, function, and variable whose type is known is enclosed in a special type function that gives the type of term. For example, the special type functions would be *point*, *set*, and *collection*, and the atom *element*(y, *sigma*(T)) would be written as *element*(*point*(y), *set*(*sigma* (*collection*(X)))). This technique enables one to simulate some sorted logics in an unsorted system. (It can also handle hierarchies of types.) See [11] for details.
3. Use implicit typing. With this technique, the position of an argument determines its type. A type is associated with each constant and function symbol, and a type is associated with each argument position of each predicate and function symbol. The element relation is partitioned into two relations: one relation, called el_p, is about points being elements of sets, and the second relation, called el_s, is about sets being elements of collections. The relations subset and equal, and the functions intersection and union are handled similarly. If input clauses have consistent use of variables (for example, there is no clause P(x) | Q(x), where the arguments of P and Q are of different types (then resolution never instantiates a variable to a term of the wrong type. Paramodulation, however, can cause a bad instantiation if paramodulation is from or into a variable.

We selected the third option for our initial experiments and for presentation of the clauses. The first option was unacceptable because the extra literals require too many extra operations to eliminate them. In addition, the first option does not prevent unwanted unifications, as do the other two options. The second option is undesirable because the extra type functions are particularly inconvenient and clumsy in this application.

The third option requires additional definitions and lemmas for the set theory concepts – for example, separate definitions and lemmas are needed for subset_s and subset_c. (The second option requires more than one version of some lemmas – for example the lemma $X \cap \emptyset = \emptyset$ would be included as set(intersect-(set(x), set(∅))) = set(∅) and collection(intersect(collection(x), collection(∅))) = collection(∅) – but only one version of each definition is required.)

The following replacements were made for the third option.

∈	el_p(point,set), el_s(set,collection)
=	eq_p(point,point), eq_s(set,set), eq_c(collection,collection)
⊆	subset_s(set,set), subset_c(collection,collection)
disjoint	disjoint_s(set,set), disjoint_c(collection,collection)
∪	Union_s(set,set), Union_c(collection,collection)
∩	inter_s(set,set), inter_c(collection,collection)
−	relative_complement_s(set,set), relative_complement_c(collection,collection)
∅	0_s, 0_c

In the following clauses, variables start with a symbol in [u, v, w, x, y, z] (lower case). The symbols [f1, f2, ..., g1, g2, ...] are Skolem functions.

Sigma (union of members)

```
1.  -el_p(u,sigma(vF)) | el_p(u,f1(vF,u))
2.  -el_p(u,sigma(vF)) | el_s(f1(vF,u),vF)
3.  el_p(u,sigma(vF)) | -el_p(u,uu1) | -el_s(uu1,vF)
```

Pi (intersection of members)

```
4.  -el_p(u,pi(vF)) | -el_s(vA,vF) | el_p(u,vA)
5.  el_p(u,pi(vF)) | el_s(f2(vF,u),vF)
6.  el_p(u,pi(vF)) | -el_p(u,f2(vF,u))
```

Topological space

```
7.  -top_space(x,vT) | eq_s(sigma(vT),x)
8.  -top_space(x,vT) | el_s(0_s,vT)
9.  -top_space(x,vT) | el_s(x,vT)
10.  -top_space(x,vT) | -el_s(y,vT) | -el_s(z,vT) | el_s(inter_s(y,z),vT)
11.  -top_space(x,vT) | -subset_c(vF,vT) | el_s(sigma(vF),vT)
12.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       el_s(f3(x,vT),vT) | subset_c(f5(x,vT),vT)
13.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       el_s(f3(x,vT),vT) | -el_s(sigma(f5(x,vT)),vT)
14.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       el_s(f4(x,vT),vT) | subset_c(f5(x,vT),vT)
15.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       el_s(f4(x,vT),vT) | -el_s(sigma(f5(x,vT)),vT)
16.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       -el_s(inter_s(f3(x,vT),f4(x,vT)),vT) | subset_c(f5(x,vT),vT)
17.  top_space(x,vT) | -eq_s(sigma(vT),x) | -el_s(0_s,vT) | -el_s(x,vT) |
       -el_s(inter_s(f3(x,vT),f4(x,vT)),vT) | -el_s(sigma(f5(x,vT)),vT)
```

Open set

```
18.  -open(u,x,vT) | top_space(x,vT)
19.  -open(u,x,vT) | el_s(u,vT)
20.  open(u,x,vT) | -top_space(x,vT) | -el_s(u,vT)
```

Closed set

```
21.  -closed(u,x,vT) | top_space(x,vT)
22.  -closed(u,x,vT) | open(rel_comp_s(u,x),x,vT)
23.  closed(u,x,vT) | -top_space(x,vT) | -open(rel_comp_s(u,x),x,vT)
```

Finer topology

```
24.  -finer(vT,vS,x) | top_space(x,vT)
25.  -finer(vT,vS,x) | top_space(x,vS)
26.  -finer(vT,vS,x) | subset_c(vS,vT)
27.  finer(vT,vS,x) | -top_space(x,vT) | -top_space(x,vS) | -subset_c(vS,vT)
```

Basis for a topology

```
28. -basis(x,vF) | eq_s(sigma(vF),x)
29. -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) |
      -el_p(y,inter_s(vB1,vB2)) | el_p(y,f6(x,vF,y,vB1,vB2))
30. -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) |
      -el_p(y,inter_s(vB1,vB2)) | el_s(f6(x,vF,y,vB1,vB2),vF)
31. -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) |
      -el_p(y,inter_s(vB1,vB2)) | subset_s(f6(x,vF,y,vB1,vB2),inter_s(vB1,vB2))
32. basis(x,vF) | -eq_s(sigma(vF),x) | el_p(f7(x,vF),x)
33. basis(x,vF) | -eq_s(sigma(vF),x) | el_s(f8(x,vF),vF)
34. basis(x,vF) | -eq_s(sigma(vF),x) | el_s(f9(x,vF),vF)
35. basis(x,vF) | -eq_s(sigma(vF),x) | el_p(f7(x,vF),inter_s(f8(x,vF),f9(x,vF)))
36. basis(x,vF) | -eq_s(sigma(vF),x) | -el_p(f7(x,vF),uu9) | -el_s(uu9,vF) |
      -subset_s(uu9,inter_s(f8(x,vF),f9(x,vF)))
```

Topology generated by a basis

```
37. -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_p(x,f10(vF,u,x))
38. -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_s(f10(vF,u,x),vF)
39. -el_s(u,top_of_basis(vF)) | -el_p(x,u) | subset_s(f10(vF,u,x),u)
40. el_s(u,top_of_basis(vF)) | el_p(f11(vF,u),u)
41. el_s(u,top_of_basis(vF)) | -el_p(f11(vF,u),uu11) | -el_s(uu11,vF) | -subset_s(uu11,u)
```

Subspace topology

```
42. -el_s(u,subspace_top(x,vT,y)) | top_space(x,vT)
43. -el_s(u,subspace_top(x,vT,y)) | subset_s(y,x)
44. -el_s(u,subspace_top(x,vT,y)) | el_s(f12(x,vT,y,u),vT)
45. -el_s(u,subspace_top(x,vT,y)) | eq_s(u,inter_s(y,f12(x,vT,y,u)))
46. el_s(u,subspace_top(x,vT,y)) | -top_space(x,vT) | -subset_s(y,x) |
      -el_s(uu12,vT) | -eq_s(u,inter_s(y,uu12))
```

Interior of a set

```
47. -el_p(u,interior(y,x,vT)) | top_space(x,vT)
48. -el_p(u,interior(y,x,vT)) | subset_s(y,x)
49. -el_p(u,interior(y,x,vT)) | el_p(u,f13(y,x,vT,u))
50. -el_p(u,interior(y,x,vT)) | subset_s(f13(y,x,vT,u),y)
51. -el_p(u,interior(y,x,vT)) | open(f13(y,x,vT,u),x,vT)
52. el_p(u,interior(y,x,vT)) | -top_space(x,vT) | -subset_s(y,x) | -el_p(u,uu13) |
      -subset_s(uu13,y) | -open(uu13,x,vT)
```

Closure of a set

```
53. -el_p(u,closure(y,x,vT)) | top_space(x,vT)
54. -el_p(u,closure(y,x,vT)) | subset_s(y,x)
55. -el_p(u,closure(y,x,vT)) | -subset_s(y,v) | -closed(v,x,vT) | el_p(u,v)
56. el_p(u,closure(y,x,vT)) | -top_space(x,vT) | -subset_s(y,x) | subset_s(y,f14(y,x,vT,u))
57. el_p(u,closure(y,x,vT)) | -top_space(x,vT) | -subset_s(y,x) | closed(f14(y,x,vT,u),x,vT)
58. el_p(u,closure(y,x,vT)) | -top_space(x,vT) | -subset_s(y,x) | -el_p(u,f14(y,x,vT,u))
```

Neighborhood

```
59. -neighborhood(u,y,x,vT) | top_space(x,vT)
60. -neighborhood(u,y,x,vT) | open(u,x,vT)
61. -neighborhood(u,y,x,vT) | el_p(y,u)
62. neighborhood(u,y,x,vT) | -top_space(x,vT) | -open(u,x,vT) | -el_p(y,u)
```

Limit point

```
63. -limit_pt(z,y,x,vT) | top_space(x,vT)
64. -limit_pt(z,y,x,vT) | subset_s(y,x)
65. -limit_pt(z,y,x,vT) | -neighborhood(u,z,x,vT) | el_p(f15(z,y,x,vT,u),inter_s(u,y))
66. -limit_pt(z,y,x,vT) | -neighborhood(u,z,x,vT) | -eq_p(f15(z,y,x,vT,u),z)
67. limit_pt(z,y,x,vT) | -top_space(x,vT) | -subset_s(y,x) |
      neighborhood(f16(z,y,x,vT),z,x,vT)
68. limit_pt(z,y,x,vT) | -top_space(x,vT) | -subset_s(y,x) |
      -el_p(uu16,inter_s(f16(z,y,x,vT),y)) | eq_p(uu16,z)
```

Boundary of a set

```
69. -el_p(u,boundary(y,x,vT)) | top_space(x,vT)
70. -el_p(u,boundary(y,x,vT)) | el_p(u,closure(y,x,vT))
71. -el_p(u,boundary(y,x,vT)) | el_p(u,closure(rel_comp_s(y,x),x,vT))
72. el_p(u,boundary(y,x,vT)) | -top_space(x,vT) | -el_p(u,closure(y,x,vT)) |
      -el_p(u,closure(rel_comp_s(y,x),x,vT))
```

Hausdorff space

```
73. -hausdorff(x,vT) | top_space(x,vT)
74. -hausdorff(x,vT) | -el_p(x_1,x) | -el_p(x_2,x) | eq_p(x_1,x_2) |
      neighborhood(f17(x,vT,x_1,x_2),x_1,x,vT)
75. -hausdorff(x,vT) | -el_p(x_1,x) | -el_p(x_2,x) | eq_p(x_1,x_2) |
      neighborhood(f18(x,vT,x_1,x_2),x_2,x,vT)
76. -hausdorff(x,vT) | -el_p(x_1,x) | -el_p(x_2,x) | eq_p(x_1,x_2) |
      disjoint_s(f17(x,vT,x_1,x_2),f18(x,vT,x_1,x_2))
77. hausdorff(x,vT) | -top_space(x,vT) | el_p(f19(x,vT),x)
78. hausdorff(x,vT) | -top_space(x,vT) | el_p(f20(x,vT),x)
79. hausdorff(x,vT) | -top_space(x,vT) | -eq_p(f19(x,vT),f20(x,vT))
80. hausdorff(x,vT) | -top_space(x,vT) | -neighborhood(uu19,f19(x,vT),x,vT) |
      -neighborhood(uu20,f20(x,vT),x,vT) | -disjoint_s(uu19,uu20)
```

Separation in a topological space

```
81. -separation(vA1,vA2,x,vT) | top_space(x,vT)
82. -separation(vA1,vA2,x,vT) | -eq_s(vA1,0_s)
83. -separation(vA1,vA2,x,vT) | -eq_s(vA2,0_s)
84. -separation(vA1,vA2,x,vT) | el_s(vA1,vT)
85. -separation(vA1,vA2,x,vT) | el_s(vA2,vT)
86. -separation(vA1,vA2,x,vT) | eq_s(Union_s(vA1,vA2),x)
87. -separation(vA1,vA2,x,vT) | disjoint_s(vA1,vA2)
88. separation(vA1,vA2,x,vT) | -top_space(x,vT) | eq_s(vA1,0_s) | eq_s(vA2,0_s) |
      -el_s(vA1,vT) | -el_s(vA2,vT) | -eq_s(Union_s(vA1,vA2),x) | -disjoint_s(vA1,vA2)
```

Connected topological space

```
89. -connected(x,vT) | top_space(x,vT)
90. -connected(x,vT) | -separation(vA1,vA2,x,vT)
91. connected(x,vT) | -top_space(x,vT) | separation(f21(x,vT),f22(x,vT),x,vT)
```

Connected set

```
92. -connected_set(vA,x,vT) | top_space(x,vT)
93. -connected_set(vA,x,vT) | subset_s(vA,x)
94. -connected_set(vA,x,vT) | connected(vA,subspace_top(x,vT,vA))
95. connected_set(vA,x,vT) | -top_space(x,vT) | -subset_s(vA,x) |
      -connected(vA,subspace_top(x,vT,vA))
```

Open covering

```
96. -open_cover(vF,x,vT) | top_space(x,vT)
97. -open_cover(vF,x,vT) | subset_c(vF,vT)
98. -open_cover(vF,x,vT) | eq_s(sigma(vF),x)
99. open_cover(vF,x,vT) | -top_space(x,vT) | -subset_c(vF,vT) | -eq_s(sigma(vF),x)
```

Compact topological space

```
100. -compact(x,vT) | top_space(x,vT)
101. -compact(x,vT) | -open_cover(vF1,x,vT) | finite(f23(x,vT,vF1))
102. -compact(x,vT) | -open_cover(vF1,x,vT) | subset_c(f23(x,vT,vF1),vF1)
103. -compact(x,vT) | -open_cover(vF1,x,vT) | open_cover(f23(x,vT,vF1),x,vT)
104. compact(x,vT) | -top_space(x,vT) | open_cover(f24(x,vT),x,vT)
105. compact(x,vT) | -top_space(x,vT) | -finite(uu24) | -subset_c(uu24,f24(x,vT)) |
       -open_cover(uu24,x,vT)
```

Compact set

```
106. -compact_set(vA,x,vT) | top_space(x,vT)
107. -compact_set(vA,x,vT) | subset_s(vA,x)
108. -compact_set(vA,x,vT) | compact(vA,subspace_top(x,vT,vA))
109. compact_set(vA,x,vT) | -top_space(x,vT) | -subset_s(vA,x) |
       -compact(vA,subspace_top(x,vT,vA))
```

5. Denials of Lemmas in Clause Form

In the following, X, T, U, Y, A, B, ... are Skolem constants.

1. The topology generated by a basis gives rise to a topological space. Denial:

```
110. basis(X,T)
111. -top_space(26,top_of_basis(T))
```

2. If (X, T) is a topological space, A is a subset of X, and every point in A has a neighborhood U that is a subset of A, then A is open in (X, T). Denial:

```
112. top_space(X,T)
113. subset_s(A,X)
114. -el_p(y,A) | neighborhood(f30(y),y,X,T)
115. -el_p(y,A) | subset_s(f30(y),A)
116. -open(A,X,T)
```

3. The subspace topology gives rise to a topological space. Denial:

```
117. top_space(X,T)
118. subset_s(Y,X)
119. -top_space(Y,subspace_top(X,T,Y))
```

4. If Y is open in X, and A is open in Y, then A is open in X. Denial:

```
120. open(Y,X,T)
121. open(A,Y,subspace_top(X,T,Y))
122. -open(A,X,T)
```

5. A finer topology induces a finer subspace topology. Denial:

```
123. finer(T1,T2,X)
124. subset_s(A,X)
125. -finer(subspace_top(X,T1,A),subspace_top(X,T2,A),X)
```

6. An alternative definition of top_of_basis. Denial:

```
126. el_p(U,top_of_basis(F)) | subset_c(G,F)
127. el_p(U,top_of_basis(F)) | eq_s(U,sigma(G))
128. -el_p(U,top_of_basis(F)) | -subset_c(x,F) | -eq_s(U,sigma(x))
```

7. Arbitrary intersections and finite unions of closed sets are closed. Denial:

```
129.  top_space(X,T)
130.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y1,X,T) | subset_s(sigma(F),X)
131.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y1,X,T) | -el_s(v,F) | closed(v,X,T)
132.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y1,X,T) | -closed(pi(F),X,T)
133.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y2,X,T) | subset_s(sigma(F),X)
134.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y2,X,T) | -el_s(v,F) | closed(v,X,T)
135.  -closed(0_s,X,T) | -closed(X,X,T) | closed(Y2,X,T) | -closed(pi(F),X,T)
136.  -closed(0_s,X,T) | -closed(X,X,T) | -closed(Union_s(Y1,Y2),X,T) | subset_s(sigma(F),X)
137.  -closed(0_s,X,T) | -closed(X,X,T) | -closed(Union_s(Y1,Y2),X,T) | -el_s(v,F) | closed(v,X,T)
138.  -closed(0_s,X,T) | -closed(X,X,T) | -closed(Union_s(Y1,Y2),X,T) | -closed(pi(F),X,T)
```

8. The interior of A is a subset of A, which is a subset of the closure of A. Denial:

```
139.  top_space(X,T)
140.  subset_s(A,X)
141.  -subset_s(interior(A,X,T),A) | -subset_s(A,closure(A,X,T))
```

9. If A is open, the interior of A is A, and if A is closed, the closure of A is A. Denial:

```
142.  top_space(X,T)
143.  subset_s(A,X)
144.  open(A,X,T) | eq_s(A,interior(A,X,T)) | closed(A,X,T) | eq_s(A,closure(A,X,T))
145.  open(A,X,T) | eq_s(A,interior(A,X,T)) | -closed(A,X,T) | -eq_s(A,closure(A,X,T))
146.  -open(A,X,T) | -eq_s(A,interior(A,X,T)) | closed(A,X,T) | eq_s(A,closure(A,X,T))
147.  -open(A,X,T) | -eq_s(A,interior(A,X,T)) | -closed(A,X,T) | -eq_s(A,closure(A,X,T))
```

10. The interior and the boundary of a set are disjoint. Denial:

```
148.  top_space(X,T)
149.  subset_s(A,X)
150.  -eq_s(inter_s(interior(A,X,T),boundary(A,X,T)),0_s)
```

11. The union of the interior and the boundary is the closure. Denial:

```
151.  top_space(X,T)
152.  subset_s(A,X)
153.  -eq_s(Union_s(interior(A,X,T),boundary(A,X,T)),closure(A,X,T))
```

12. If the boundary of A is empty, A is both open and closed. Denial:

```
154.  top_space(X,T)
155.  subset_s(A,X)
156.  eq_s(boundary(A,X,T),0_s) | open(A,X,T)
157.  eq_s(boundary(A,X,T),0_s) | closed(A,X,T)
158.  -eq_s(boundary(A,X,T),0_s) | -open(A,X,T) | -closed(A,X,T)
```

13. If some limit points are added to a connected set, the result is connected. Denial:

```
159.  connected_set(A,X,T)
160.  -el_p(y,B) | limit_pt(y,A,X,T)
161.  -connected_set(Union_s(A,B),X,T)
```

14. The closure of a connected set is connected. Denial:

```
162.  connected_set(A,X,T)
163.  -connected_set(closure(A,X,T),X,T)
```

6. A Proof of Lemma 1 in Five Parts

This section contains resolution (with factoring) proofs of Lemma 1. The purpose of our presentation of these proofs is not to demonstrate that our theorem prover can find the proofs, but rather to simply show some example resolution proofs and the types of clause and term that appear in them. The lemma asserts that the topology generated by a basis is in fact a topological space. It has been divided into five parts: one to prove each of the five conditions necessary for a topological space. Proof outlines were first worked out by hand, then just the clauses required for refutations were given to the automated theorem prover OTTER [6]. After several attempts at each part, modifying the parameters and options, the following proofs were found by OTTER within a few seconds. (See Section 7 for comments about this approach.)

The symbols [F, X, U, V] are Skolem constants. Clauses for the set theory axioms, definitions, and lemmas that do not appear in Section 4 are marked with an asterisk (*).

Proof of Lemma 1a:

$$\forall F\, \forall X(basis(X, F) \rightarrow sigma(top\_of\_basis(F)) \subseteq X)$$

```
1 -basis(x,vF) | eq_s(sigma(vF),x).
2 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_p(x,f10(vF,u,x)).
3 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_s(f10(vF,u,x),vF).
4 -el_p(u,sigma(vF)) | el_p(u,f1(u,vF)).
5 -el_p(u,sigma(vF)) | el_s(f1(u,vF),vF).
6 el_p(u,sigma(vF)) | -el_p(u,uu1) | -el_s(uu1,vF).
7* subset_s(x,x).
8* -subset_s(x,y) | -el_p(u,x) | el_p(u,y).
9* -eq_s(x,y) | subset_s(x,y).
10* subset_s(x,y) | el_p(g1(x,y),x).
11* subset_s(x,y) | -el_p(g1(x,y),y).
12 basis(X,F).
13 -subset_s(sigma(top_of_basis(F)),X).
14 (12,1) eq_s(sigma(F),X).
15 (14,9) subset_s(sigma(F),X).
17 (13,11) -el_p(g1(sigma(top_of_basis(F)),X),X).
18 (13,10) el_p(g1(sigma(top_of_basis(F)),X),sigma(top_of_basis(F))).
20 (17,8) -subset_s(x,X) | -el_p(g1(sigma(top_of_basis(F)),X),x).
23 (18,5) el_s(f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F)),top_of_basis(F)).
24 (18,4) el_p(g1(sigma(top_of_basis(F)),X),f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F))).
27 (24,8) -subset_s(f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F)),x) |
       el_p(g1(sigma(top_of_basis(F)),X),x).
32 (23,3) -el_p(x,f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F))) |
       el_s(f10(F,f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F)),x),F).
33 (23,2) -el_p(x,f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F))) |
       el_p(x,f10(F,f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F)),x)).
35 (20,15) -el_p(g1(sigma(top_of_basis(F)),X),sigma(F)).
53 (35,6) -el_p(g1(sigma(top_of_basis(F)),X),x) | -el_s(x,F).
76 (32,27,7) el_s(f10(F,f1(g1(sigma(top_of_basis(F)),X),top_of_basis(F)),
       g1(sigma(top_of_basis(F)),X)),F).
85 (33,27,7) el_p(g1(sigma(top_of_basis(F)),X),f10(F,f1(g1(sigma(top_of_basis(F)),X),
       top_of_basis(F)),g1(sigma(top_of_basis(F)),X))).
164 (53,85,76) .
```

Proof of Lemma 1b:

$$\forall F\, \forall X(basis(X, F) \rightarrow 0 \in top\_of\_basis(F))$$

```
2 el_s(u,top_of_basis(vF)) | el_p(f11(vF,u),u).
10* -el_p(x,0_s).
14 -el_s(0_s,top_of_basis(F)).
17 (14,2,10) .
```

Proof of Lemma 1c:

$$\forall F\, \forall X(basis(X, F) \rightarrow X \in top\_of\_basis(F))$$

```
1 -basis(x,vF) | eq_s(sigma(vF),x).
2 el_s(u,top_of_basis(vF)) | el_p(f11(vF,u),u).
3 el_s(u,top_of_basis(vF)) | -el_p(f11(vF,u),uu11) | -el_s(uu11,vF) | -subset_s(uu11,u).
4 -el_p(u,sigma(vF)) | el_p(u,f1(vF,u)).
5 -el_p(u,sigma(vF)) | el_s(f1(vF,u),vF).
7* -el_s(x,y) | subset_s(x,sigma(y)).
8* -subset_s(x,y) | -el_p(u,x) | el_p(u,y).
9* subset_s(x,x).
10* -eq_s(x,y) | -subset_s(z,x) | subset_s(z,y).
11* -eq_s(x,y) | -subset_s(x,z) | subset_s(y,z).
12 basis(X,F).
13 -el_s(X,top_of_basis(F)).
14 (12,1) eq_s(sigma(F),X).
16 (13,2) el_p(f11(F,X),X).
17 (14,11) -subset_s(sigma(F),x) | subset_s(X,x).
18 (14,10) -subset_s(x,sigma(F)) | subset_s(x,X).
24 (17,9) subset_s(X,sigma(F)).
31 (24,8) -el_p(x,X) | el_p(x,sigma(F)).
36 (18,7) subset_s(x,X) | -el_s(x,F).
47 (36,3) -el_s(x,F) | el_s(X,top_of_basis(y)) | -el_p(f11(y,X),x) | -el_s(x,y).
49 (47,13) -el_s(x,F) | -el_p(f11(F,X),x).
67 (31,16) el_p(f11(F,X),sigma(F)).
76 (67,5) el_s(f1(F,f11(F,X)),F).
77 (67,4) el_p(f11(F,X),f1(F,f11(F,X))).
122 (49,76,77) .
```

Proof of Lemma 1d:

$$\forall F\, \forall X\, \forall U\, \forall V\left(\begin{bmatrix} basis(X, F) \wedge \\ U \in top\_of\_basis(F) \wedge \\ V \in top\_of\_basis(F) \end{bmatrix} \rightarrow (U \cap V) \in top\_of\_basis(F)\right)$$

```
1 -basis(x,vF) | eq_s(sigma(vF),x).
2 -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) | -el_p(y,inter_s(vB1,vB2)) |
        el_p(y,f6(x,vF,y,vB1,vB2)).
3 -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) | -el_p(y,inter_s(vB1,vB2)) |
        el_s(f6(x,vF,y,vB1,vB2),vF).
4 -basis(x,vF) | -el_p(y,x) | -el_s(vB1,vF) | -el_s(vB2,vF) | -el_p(y,inter_s(vB1,vB2)) |
        subset_s(f6(x,vF,y,vB1,vB2),inter_s(vB1,vB2)).
5 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_p(x,f10(vF,u,x)).
6 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_s(f10(vF,u,x),vF).
7 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | subset_s(f10(vF,u,x),u).
8 el_s(u,top_of_basis(vF)) | el_p(f11(vF,u),u).
9 el_s(u,top_of_basis(vF)) | -el_p(f11(vF,u),uu11) | -el_s(uu11,vF) | -subset_s(uu11,u).
10* -subset_s(x,y) | -subset_s(y,z) | subset_s(x,z).
11* -el_p(z,inter_s(x,y)) | el_p(z,x).
12* -el_p(z,inter_s(x,y)) | el_p(z,y).
13* el_p(z,inter_s(x,y)) | -el_p(z,x) | -el_p(z,y).
14* -subset_s(x,y) | -subset_s(u,v) | subset_s(inter_s(x,u),inter_s(y,v)).
15 -el_s(u,x) | -el_p(z,u) | el_p(z,sigma(x)).
16* -eq_s(x,y) | -el_p(z,x) | el_p(z,y).
17 basis(X,F).
18 el_s(U,top_of_basis(F)).
19 el_s(V,top_of_basis(F)).
```

```
20 -el_s(inter_s(U,V),top_of_basis(F)).
22* eq_s(inter_s(x,y),inter_s(y,x)).
29 (17,1) eq_s(sigma(F),X).
72 (20,8) el_p(f11(F,inter_s(U,V)),inter_s(U,V)).
76 (72,12) el_p(f11(F,inter_s(U,V)),V).
77 (72,11) el_p(f11(F,inter_s(U,V)),U).
105 (76,7,19) subset_s(f10(F,V,f11(F,inter_s(U,V))),V).
106 (76,6,19) el_s(f10(F,V,f11(F,inter_s(U,V))),F).
107 (76,5,19) el_p(f11(F,inter_s(U,V)),f10(F,V,f11(F,inter_s(U,V)))).
121 (77,7,18) subset_s(f10(F,U,f11(F,inter_s(U,V))),U).
122 (77,6,18) el_s(f10(F,U,f11(F,inter_s(U,V))),F).
123 (77,5,18) el_p(f11(F,inter_s(U,V)),f10(F,U,f11(F,inter_s(U,V)))).
155 (107,15,106) el_p(f11(F,inter_s(U,V)),sigma(F)).
163 (121,14,105,22,22) subset_s(inter_s(f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V)))),inter_s(U,V)).
191 (123,13,107,22) el_p(f11(F,inter_s(U,V)),inter_s(f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V))))).
208 (155,16,29) el_p(f11(F,inter_s(U,V)),X).
317 (191,4,17,208,122,106) subset_s(f6(X,F,f11(F,inter_s(U,V)),f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V)))),inter_s(f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V))))).
318 (191,3,17,208,122,106) el_s(f6(X,F,f11(F,inter_s(U,V)),f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V)))),F).
319 (191,2,17,208,122,106) el_p(f11(F,inter_s(U,V)),f6(X,F,f11(F,inter_s(U,V)),
      f10(F,U,f11(F,inter_s(U,V))),f10(F,V,f11(F,inter_s(U,V))))).
337 (319,9,20,318) -subset_s(f6(X,F,f11(F,inter_s(U,V)),f10(F,U,f11(F,inter_s(U,V))),
      f10(F,V,f11(F,inter_s(U,V)))),inter_s(U,V)).
487 (337,10,317,163) .
```

Proof of Lemma 1e:

$$\forall F\,\forall X\,\forall G\left(\begin{bmatrix} basis(X,F) \wedge \\ G \subseteq top\_of\_basis(F) \end{bmatrix} \rightarrow sigma(G) \in top\_of\_basis(F)\right)$$

```
1 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_p(x,f10(vF,u,x)).
2 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | el_s(f10(vF,u,x),vF).
3 -el_s(u,top_of_basis(vF)) | -el_p(x,u) | subset_s(f10(vF,u,x),u).
4 el_s(u,top_of_basis(vF)) | el_p(f11(vF,u),u).
5 el_s(u,top_of_basis(vF)) | -el_p(f11(vF,u),uu11) | -el_s(uu11,vF) | -subset_s(uu11,u).
6 -el_p(u,sigma(vF)) | el_p(u,f1(vF,u)).
7 -el_p(u,sigma(vF)) | el_s(f1(vF,u),vF).
8* subset_s(x,y) | -el_p(u,x) | el_p(u,y).
9* -subset_s(x,y) | -el_s(y,z) | subset_s(x,sigma(z)).
10* -subset_c(x,y) | -el_s(u,x) | el_s(u,y).
12 subset_c(G,top_of_basis(F)).
13 -el_s(sigma(G),top_of_basis(F)).
17 (13,4) el_p(f11(F,sigma(G)),sigma(G)).
19 (17,7) el_s(f1(G,f11(F,sigma(G))),G).
20 (17,6) el_p(f11(F,sigma(G)),f1(G,f11(F,sigma(G)))).
25 (19,10) -subset_c(G,x) | el_s(f1(G,f11(F,sigma(G))),x).
33 (25,12) el_s(f1(G,f11(F,sigma(G))),top_of_basis(F)).
43 (33,3) -el_p(x,f1(G,f11(F,sigma(G)))) | subset_s(f10(F,f1(G,f11(F,sigma(G))),x),
      f1(G,f11(F,sigma(G)))).
44 (33,2) -el_p(x,f1(G,f11(F,sigma(G)))) | el_s(f10(F,f1(G,f11(F,sigma(G))),x),F).
45 (33,1) -el_p(x,f1(G,f11(F,sigma(G)))) | el_p(x,f10(F,f1(G,f11(F,sigma(G))),x)).
57 (43,8) subset_s(f10(F,f1(G,f11(F,sigma(G))),x),f1(G,f11(F,sigma(G)))) |
      subset_s(y,f1(G,f11(F,sigma(G)))) | -el_p(x,y).
61 (57) subset_s(f10(F,f1(G,f11(F,sigma(G))),x),f1(G,f11(F,sigma(G)))) |
      -el_p(x,f10(F,f1(G,f11(F,sigma(G))),x)).
65 (44,20) el_s(f10(F,f1(G,f11(F,sigma(G))),f11(F,sigma(G))),F).
75 (45,20) el_p(f11(F,sigma(G)),f10(F,f1(G,f11(F,sigma(G))),f11(F,sigma(G)))).
84 (75,5,13,65) -subset_s(f10(F,f1(G,f11(F,sigma(G))),f11(F,sigma(G))),sigma(G)).
88 (84,9) -subset_s(f10(F,f1(G,f11(F,sigma(G))),f11(F,sigma(G))),x) | -el_s(x,G).
110 (88,61,19,75) .
```

7. Conclusion

The lemmas in Section 3 are not difficult for student mathematicians. Most can be proved by using the topology definitions to open up the defined concepts in the lemmas, then using basic set theory to complete the proofs. One of the reasons the lemmas are combinatorially difficult for resolution-style theorem provers is that conversion to clause form causes much replication of literals. An obvious strategy to reduce literal replication in the denial of the conclusion is to break a problem into independent subproblems before converting to clauses, as in the five-part proof of Lemma 1. But even with reduction to independent subproblems, we suspect that the basic representation and framework used by OTTER is inadequate for problems of his kind. A variant of linked inference rules [12] or of non-clausal theorem proving [1, 10, 8] might be effective.

We have avoided the issue of how to deal with the underlying basic set theory because of our wish to focus on topology concepts. Typing of objects may make these problems easier to handle than problems in other areas built on set theory; it enables the use of naive set theory rather than the full set theory presented in [2].

Acknowledgement

We thank Farshid Ketabchi for suggesting several improvements and for pointing out several important errors in the definitions.

References

1 Bledsoe, W. W., 'Non-resolution theorem proving', *Artificial Intelligence* **9**, 1–35 (1977).
2. Boyer, R., Lusk, E., McCune, W., Overbeek, R., Stickel, M., and Wos, L., 'Set theory in first-order logic: Clauses for Gödel's axioms', *J. Automated Reasoning* **2**, 287–327 (1986).
3. Gödel, K., *The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of Set Theory*, Princeton University Press (1940).
4. Kelley, J. L., *General Topology*, D. Van Nostrand, New York (1955).
5. Lusk, E. and Overbeek, R., 'The automated reasoning system ITP', ANL-84-27, Argonne National Laboratory, Argonne, Ill. (April 1984).
6. McCune, W., 'OTTER user's guide', ANL-88-44, Argonne National Laboratory, Argonne, Ill. (1988).
7 Munkres, J. R., *Topology: A First Course*, Prentice-Hall, Englewood Cliffs, N.J. (1975).
8. Potter, R. C. and Plaisted, D. A., 'Term rewriting: some experimental results', *Proceedings of the 9th International Conference on Automated Deduction*, Springer-Verlag Lecture Notes in Computer Science 310, 435–454 (1988).
9. Smith, B. T., 'Reference manual for the environmental theorem prover, an incarnation of AURA', ANL-88-2, Argonne National Laboratory, Argonne, IL (1988).
10. Stickel, M., 'A non-clausal connection-graph resolution theorem-proving program', *Proceedings of the AIAA-82 National Conference on Artificial Intelligence*, Pittsburgh, 229–233 (1982).
11. Stickel, M., 'Schubert's steamroller problem: formulations and solutions', *J. Automated Reasoning* **2**, 89–101 (1986).
12. Wos, L., Veroff, R., Smith, B. T., and McCune, W., 'The linked inference principle II: the user's view', *Proceedings of the Seventh International Conference on Automated Deduction*, Springer-Verlag Lecture Notes in Computer Science 170, 316–332 (1984).
13 Wos, L., Overbeek, R., Lusk, E., and Boyle, J., *Automated Reasoning: Introduction and Applications*, Prentice-Hall, Englewood Cliffs, N.J. (1984).