A Partially Automated Proof of the Cantor-Bernstein Theorem

Ian Kash, iak@andrew.cmu.edu

April 30, 2004

Abstract

AProS is a complete search procedure for first order logic. This work extends AProS to reason with definitions and axioms in order to perform mathematical proof search. Set theory was selected as a test of the system, with the particular goal of proving the Cantor-Bernstein Theorem. Automated proofs of two important lemmas and several other theorems are presented, along with a discussion of the heuristic guidance necessary for proving them. A partially automated proof of the theorem is presented with a discussion of the difficulties in achieving a fully automated proof.

1 Introduction

AProS (Automated Proof Search) is a complete search procedure for first order logic. The major goal of the AProS project is to efficiently produce natural proofs that resemble those given by a human presented with the same problem. An advantage of such proofs is that they have structure so that a human examining them can see how the problem is broken down and gain some insight from the solution.

Previous work has demonstrated the flexibility of the AProS architecture, adapting it to intuitionistic logic [3] and extending it to prove Gödel's incompleteness theorems at an abstract axiomatic level; see [4]. I wanted to extend AProS to more standard mathematical reasoning. The particular domain I decided to work with was set theory. A major motivating factor is that set theory has a large number of axioms and defined concepts. If there were only a few, their effect on the number of options available to the search would be relatively small and their implementation could be relatively inefficient. The larger number not only forces an efficient implementation, but seems to provide enough options to require real heuristic guidance to make the search possible. Even though not all of the axioms and basic definitions of ZF are actually used, the subset required is still large enough to pose a challenge.

2 The Cantor-Bernstein Theorem

The Cantor-Bernstein theorem states that given sets X and Y and injections $i : X \to Y$ and $j : Y \to X$ then there is a bijection $f : X \to Y$. While this is trivial in the finite case, it is not in the infinite case. The classical proof of it (exhibited in Appendix 1) involves much of the basic machinery of set theory, including unions, power-sets, functions, injections, and bijections. At the same time it avoids more complicated developments such as cardinality. Thus this theorem seems to provide a challenging, but not unreachable goal.

The approach I have adopted is outlined in [1] and is based on two lemmas. The first lemma states that every monotone function has a fixed point. Specifically, let X be a set and $h : P(X) \to P(X)$ is such that $A \subseteq B \subseteq X \to h(A) \subseteq h(B)$. Then $\exists T \subseteq P(X)$ such that $h(\bigcup T) = \bigcup T$. The second lemma states that a particular function * is monotone. Specifically, given sets X and Y and injections $i : X \to Y$ and $j : Y \to X$, let $* : P(X) \to P(X)$ be defined as $A^* = X - j[Y - i[A]]$ for each $A \subseteq X$. Then $A \subseteq B \subseteq X \to A^* \subseteq B^*$. From the lemmas we get $\exists T \subseteq P(X)$ such that $\bigcup T = (\bigcup T)^*$. Since we do not actually care that this set is a union, we will instead use the statement $\exists T \subseteq X$ such that $T = T^*$ Let $f = \begin{cases} i(x) & x \in T \\ j^{-1}(x) & x \notin T \end{cases}$. Then to prove the theorem all that remains is to show that f is a bijection.

3 AProS and Extensions

AProS is based on the intercalation calculus as described in [2] and [5]. It works by trying to “close the gap” between assumptions and the goal through applications of appropriate rules. The (inverted) *introduction* rules transform a goal into a new goal or goals sufficient to prove the current goal. For example, the $\&I$ rule says that if the goal is $A\&B$ then we can separately prove A and B in order to conclude $A\&B$. The *elimination* rules allow additions to the assumptions. For example, if the assumptions include $A\&B$, then the $\&E$ rule allows either A or B to be added to the assumptions. Rather than explicitly keeping the full set of assumptions that can be generated by these rules, AProS keeps only the base assumptions paired with the current goal in what is known as an *occurrence.* Only when it finds a sequence of these rules that bridge the gap from an assumption to one of the current goals does it apply them. Such a sequence of rules is an *extraction.* For example, $\to E, \&E$ is an extraction of B from $A \to (B\&C)$.

A full description of the machinery necessary for handling quantifiers is beyond the scope of this thesis. A description of the underlying theory can be found in [2] and a technical report describing the implementation is in progress. One concept that does bear mentioning is that of an *extraction boundary.* In the first order case, extractions may involve the unification of terms. Two extractions for two different goals may unify a particular variable with a different constant. In order to ensure that no such extractions are done, all occurrences that have extractions involving the unification of terms are added to the extraction boundary. Once every occurrence that still needs justification is in the extraction boundary, all of the extractions can be applied simultaneously in order to ensure consistency. When an extraction boundary fails because there is no consistent set of extractions, a considerable number of possibilities need to be explored in order to ensure the completeness of the search procedure. If the extraction boundary contains occurrences not part of the path to a proof, this can lead to a very deep and fruitless search and prevent a proof from being found in any reasonable amount of time. Thus an important consideration for some of the heuristics discussed later is to make good initial decisions about exactly which possibilities to explore.

In order to permit mathematical reasoning, I extended AProS to provide support for definitions and axioms. Support for definitions took the form of two new rules. DefI is a rule that allows the application of a definition to obtain a new goal. For example with the goal $X \subseteq Y$, DefI allows the new goal $(\forall z)(z \in X \rightarrow z \in Y)$. It is attempted as the last rule before indirect argument is attempted (i.e. after all purely logical possibilities for extractions and introductions have been explored). DefE is a rule that allows the same transformation to be performed in the course of an extraction. Thus the same procedure can be applied at the start when possible strategies using the assumptions are being explored.

Support for axioms does not require any additional rules; however in order to allow for different handling of extractions from axioms relative to extractions from problem specific assumptions, they are examined separately. As will be discussed later, all of the axioms for set theory are actually implemented as "definitions". Therefore the machinery for axioms is not used in practice but would be required for a different formulation.

4 Axioms and Definitions

Proving everything directly from the axioms is impractical; proofs would be extremely long and the searches would be extremely large. Thus I have taken a somewhat higher level approach. Those axioms explicitly used are implemented as definitions rather than axioms. This is very natural for the axiom of extensionality which is implemented as $X = Y \Leftrightarrow X \subseteq Y \& Y \subseteq X$. The axioms of union and powerset are usually phrased in terms of the existence of appropriate sets. However it is much more convenient and natural to treat these as operators that can be applied to a set (and the introduction of such operators is easily justified by the appropriate axiom and extensionality). Thus they are implemented as $x \in \bigcup X \Leftrightarrow (\exists Y)(Y \in X \& x \in Y)$ and $x \in P(X) \Leftrightarrow x \subseteq X$ respectively. For examples of how proofs using these work see the proof of the transitivity of subsets and the proof that $\bigcup P(X) = X$ in Appendix 2.

The remaining axioms are used only implicitly in a number of additional definitions. For example, the difference operator is defined as $x \in X - Y \Leftrightarrow x \in X \& \sim x \in Y$. The fact that this is a set relies on the axiom of comprehension. A number of other definitions specific to each problem are also used and will be discussed in the appropriate sections.

5 The First Lemma

The first lemma involves the definition of the set $T = \{A \subseteq X | A \subseteq h(A)\}$. This definition is specific to this problem, and so should not be part of the global set of definitions for set theory. One way to include it would be as an explicit premise such as $A \in T \leftrightarrow A \subseteq h(A)$. However, this results in the use of the biconditional for two different purposes. In the proof search engine, definitional statements should be handled differently from assumptions. In particular, there is no reason to attempt an extraction from a definition. However, since there was no way for AProS to draw this distinction, it attempted extractions from it which led to a very deep search; indeed, no proof of the first lemma was found in any reasonable amount of time. A better solution is to reserve the use of the biconditional for its original purpose and handle such problem specific definitions by providing a mechanism to add definitions for a specific problem. This is a very natural way to draw the needed distinction and allows the proof to go through very easily.

One additional improvement was needed to produce the proof presented in Appendix 2. This is actually an improvement made to AProS proper rather than one specific to mathematical reasoning. When an extraction boundary fails, one possibility AProS explores is to select an occurrence to try non-extraction based strategies in order to find a new extraction boundary. The original implementation just selected an arbitrary occurrence. This led to an occurrence being chosen that did not need further exploration. In this case, the only remaining strategy was to use an indirect argument. While AProS was still able to find a proof, the resulting proof had a pointless indirect argument. The new heuristic is to count the number of strategies available for each occurrence. Then the occurrence with the greatest number of strategies should be attempted. Thus occurrences that only have indirect strategies remaining are passed over in favor of ones that have other strategies as well and seem more likely to lead to a proof. This avoids the unnecessary indirect argument and yields the proof presented in Appendix 2.

However, the proof presented in Appendix 2 is still not as natural as it could be. The subproof that $\bigcup T \subseteq h(\bigcup T)$ is repeated twice. AProS does support caching of subproofs, but this currently only works when the entire first proof is completed before it is encountered again. A possible future area of research is creating a post-processing method that can identify repeated arguments such as this and replace them with a single instance.

6 The Second Lemma

The second lemma also uses two additional definitions. The first is the general notion of an image, which is defined as $x \in i[Y] \Leftrightarrow (\exists y)(y \in Y \,\&\, i(y) = x)$. The second is the definition of the function * as $x \in X^* \Leftrightarrow x \in A - j[B - i[X]]$ (where $i : A \rightarrow B$ and $j : B \rightarrow A$). Once the definitions were formulated, no additional heuristics were needed to get the proof presented in Appendix 2. Note that the statement of the lemma given in section 2 included the stipulation that i and j were injective, but no mention of injectivity is present here. As it turns out a stronger version of the lemma that does not include this requirement is also true. I did not realize this at the time and did not expect a successful proof. Thus AProS was actually able to prove a mathematical result I did not know to be true.

As with the proof of the first lemma, the proof of the second lemma is fairly natural but there are several flaws that could be addressed. The first is a minor issue. Certain portions of AProS are still nondeterministic, so the proof is actually one of the three different ones that are produced. Ideally this nondeterminism should be eliminated. The source of this is most likely the use of the Java class HashSet, which has a nondeterministic iteration order.

A flaw with the proof itself is that lines 31-41 are unnecessary and simply transform one contradiction into another. There are two avenues along which this portion of the proof could be improved. The first would be an improved heuristic for selecting contradictory pairs. The extra argument occurs because the wrong contradiction is selected as a goal. Thus if the correct contradiction were selected this would not happen. An alternative is to create a post-processing step that identifies cases where this has occurred and eliminates the unnecessary argument.

The definitions were originally formulated with the statement $i(y) = x$ being represented as a three place predicate with no actual meaning assigned to it. Once the definitions for the theorem were reformulated using the application operator (see section 8) it was natural to replace this special predicate with equality and application. This originally led to a very unnatural proof with lots of unnecessary indirect arguments. However, removing the axiom of extensionality (which effectively restored it to an undefined three place predicate) caused the original proof to return. The problem is that the definition of equality provides $DefI$ strategies for occurrences that should always be extracted in the context of this problem. Thus extraction boundary reasoning leads to extra contradictions as described in section 5.

7 A First Approach to the Theorem

My first formulation of the definitions for the proof of the theorem was based on the idea of defining functions as sets of ordered pairs. Thus injectivity was defined as $I(f) \Leftrightarrow (\forall x)(\forall y)(\sim x = y \rightarrow (\forall z)(\sim ((x,z) \in f \,\&\, (y,z) \in f)))$. As a warmup I decided to prove that the identity function defined as $(x,y) \in i \Leftrightarrow x = y$ is injective. The proof is presented in Appendix 2 and is fairly natural. Since the definition provides nothing about equality other than extensionality, the proof includes two subproofs of the transitivity of equality. Thus the proof could be improved and shortened by a good implementation of rules and axioms for identity.

At first it seemed this approach would extend very naturally to proving the injectivity of f defined as $(x,y) \in f \Leftrightarrow ((x,y) \in i \,\&\, x \in T) \vee ((y,x) \in j \,\&\, \sim x \in T)$. In examining the attempts at a proof it first appeared that a few heuristics would be needed to help make better decisions and then the proof would go through. However, as it turns out this is not the case because *any* mistake leads to a very deep series of indirect arguments that runs for over 100,000 steps without returning. Even with excellent heuristics, an occasionally wrong strategy will be explored. Demanding perfect strategic guidance is unreasonable.

This mode of failure demonstrates a serious flaw with the approach currently taken by AProS. Once a strategy is selected, it never tries any other strategy for that occurrence until all possibilities have been exhausted along that branch. Thus when the space of potential strategies is very large, this causes any mistake to render the problem intractable. It is fairly easy to create such examples. $\{(\forall x)(\forall y)((Q(x) \,\&\, R(y)) \rightarrow Q(y)), \sim P(y), (\forall x)(P(x) \vee Q(x)), (\forall x)(Q(x) \leftrightarrow R(x)), (\exists x)(P(x) \rightarrow Q(x)), \sim Q(z)\} \vdash N \vee \sim N$ has this same problem. In this case the premises have nothing to do with the desired conclusion and are noteworthy only in that they do not contain a contradiction. While this may seem artifical as a problem, subproblems that involve none of the assumptions are fairly common. In this example after 120,000 search steps AProS has not even begun to attempt an indirect argument on $N \vee \sim N$. Any method that pursues a strategy until all possibilities have been exhausted will have such problematic examples. Thus a mechanism is needed to mimic the human tendency to “give up and try something else.”

To see if a proof of the surjectivity of f could be achieved despite the problems with the injectivity, I first attempted to prove that f is a function from A to B defined as $F(f,a,b) \Leftrightarrow (\forall x)(\forall y)((x,y) \in f \rightarrow (x \in a \,\&\, y \in b)) \,\&\, (\forall x)(x \in a \rightarrow (\exists y)((x,y) \in f)) \,\&\, (\forall x)(\forall y)(\forall z)(((x,y) \in f \,\&\, (x,z) \in f) \rightarrow y = z)$, a concept which is a precondition of defining surjectivity. Not only did the definition as a whole fail in the same fashion. All three sub-arguments failed when attempted individually. Thus given the current state of AProS it appears this approach is inadequate for proving results about functions of any reasonable complexity.

8 A Second Approach to the Theorem

My second approach was to attempt to avoid some of the problems by working at a higher level of abstraction. Thus rather than viewing functions as sets of ordered pairs, I recast the definitions in terms of an apply operator that applied a function to a value, and equality. Thus injectivity was now defined as $I(f, a, b) \Leftrightarrow F(f, a, b) \,\&\, (\forall x)(\forall y)(x \in a \,\&\, y \in a) \rightarrow (f(x) = f(y) \rightarrow x = y))$ (Where $F(f, a, b)$ means f is a function from a to b). I also gave it access to a number of other facts that could be proved as lemmas. I gave it two facts about j^{-1}: $j(y) = x \Leftrightarrow j^{-1}(x) = y$ and $j^{-1}(x) = j^{-1}(y) \Leftrightarrow x \in a \,\&\, y \in a \,\&\, x = y$. I also gave it the facts that identity is symmetric ($x = y \Leftrightarrow y = x$) and that f is a function from a to b ($F(f, a, b)$). In addition I gave it the heuristic that when trying to prove something of the form $f(x) = y$ it should try or-elimination first. Since f is defined piecewise arguing by cases in this fashion is a very natural thing to do. Even at this level of abstraction, the problems described in section 7 prevent a proof. However, at this level structure can be seen that indicates that such a proof would be possible if AProS did not go off on deep searches of fruitless areas.

Looking at the steps taken for the proof that f is injective, AProS successfully sets up an attempted proof of $x = y$ into 4 cases that correspond to $(x \in t \,\&\, y \in t)$, $(\sim x \in t \,\&\, y \in t)$, $(x \in t \,\&\, \sim y \in t)$, and $(\sim x \in t \,\&\, \sim y \in t)$. After this it gets lost in a deep search. When each of the cases are attempted individually, it very easily proves the first and fourth cases by the injectivity of i and j respectively. However in the second and third case (which are really only one case because they are symmetric) it gets lost almost immediately. However if the first steps are taken for it, it finds a proof fairly quickly (with the proof being somewhat lengthy only because it again must use extensionality to argue for the transitivity of equality). The automated proof sections and the part missing that must be done by hand are in Appendix 3. A similar analysis can be applied to the surjectivity of f, by supplying Apros with the additional fact that $(\forall x)(x \in i[t] \vee \sim x \in i[t])$.

9 Concluding Remarks and Future Work

There are a number of improvements that are not critical, but would make AProS more usable for mathematical proof search. Post-processing mechanisms such as those described in sections 5 and 6 would lead to more natural proofs. Currently definitions and axioms are not naturally included in the AProS display. They are instead loaded in manually. A method for specifying a set of definitions and axioms to be used for a group of assertions would be useful. To achieve this the parser should be adapted to allow the user to input problem specific definitions and axioms, and a proper renderer should be created.

There are three important lessons to draw from this work. The first is that any algorithm that applies rules in a fixed order and never abandons a strategy until all further possibilities have been exhausted can always be drawn off into a deep and fruitless search that prevents a proof from being found in any reasonable amount of time. Furthermore such examples are not merely pathological cases but are broad classes of problems that include many real and interesting problems. Thus the single most important piece of future work is altering the search to include a mechanism for abandoning a strategy before exhausting it as described in section 7. Without this, any sufficiently complex result, whether in mathematics or first order logic, will remain out of reach. Nearly as important is improving the extraction boundary reasoning in a similar fashion. Not only can poor choices in extraction boundary reasoning lead to unnatural arguments as described in sections 5 and 6, but not backtracking before all possibilities of expansion have been explored can lead to the same problems here that it does for strategies.

The second important lesson is that a reasoned implementation of equality is essential for natural proofs. Many of the proofs presented in Appendix 2 and 3 include lengthy subproofs of the transitivity of equality. Since there is nothing available to the search other than the axiom of extensionality this is unavoidable. However it is obfuscating and unnatural to have such subproofs appear in proof after proof. A reasoned implementation of equality that naturally provided such features as transitivity could be verified once and then used from then on to produce more natural proofs.

The third important lesson is that in mathematical proof search it is important to build up layers of abstraction. Just as a mathematics textbooks will build more and more abstract machinery on top of the basic axioms to produce elegant proofs, the proof search must use ideas further and further from the axioms if natural proofs are to be produced. The second approach to the theorem described in section 8 is a good demonstration of the advantages of using some more abstract machinery. While the same issues that prevented the lower level approach from producing a proof ultimately prevented it from doing so as well, what it was able to produce is far more structured and natural and captures far more of the problem. This approach is by no means the highest level the problem can be approached at; there is definitely room for approaches using even more abstract machinery that may prove more successful even with the current state of AProS. This results when this approach is fully applied can be seen from the proofs of Gödel's incompleteness theorems in [4]. The success in producing natural proofs relies heavily on the implementation of and heuristics for higher level concepts such as the representability and derivability conditions. Without these it is hard to see how the those proofs would have been possible.

Once AProS has been improved enough to be able to truly support mathematical proof search, there are a number of other areas that could be explored. Set theory could be explored both closer to the axioms, such as justifying the use of the operators described in section 4, and with more complex concepts such as cardinality. In addition the same framework should support exploration of other areas of mathematics such as group theory or category theory.

References

[1] K. Devlin. *Fundamentals of Contemporary Set Theory.* Springer-Verlag, 1979.

[2] W. Sieg and J. Byrnes. Normal natural deduction proofs (in classical logic). *Studia Logica*, (60):67–106, 1998.

[3] W. Sieg and S. Cittadini. Normal natural deduction proofs (in non-classical logics). Technical Report CMU-PHIL-130, Carnegie Mellon University, 2002.

[4] W. Sieg and C. Field. Automated search for gödel's proofs. To appear in Annals of Pure and Applied Logic.

[5] W. Sieg and R. Scheines. Searching for proofs (in sentential logic). In L. Burkholder, editor, *Philosophy and the Computer*, pages 137–159. Westview Press, 1992.

Appendix 1: Proofs of Lemmas and Theorem by hand

Lemma 1

Let X be a set and $h : P(X) \to P(X)$ is such that $A \subseteq B \subseteq X \to h(A) \subseteq h(B)$. Then $\exists T \subseteq P(X)$ such that $h(\bigcup T) = \bigcup T$.

Let $T = \{A \subseteq X | A \subseteq h(A)\}$. Let $x \in \bigcup T$ be given. Then there is some $A \in T$ such that $x \in A$. Since $A \in T$, $A \subseteq h(A)$ and so $x \in h(A)$. $A \subseteq \bigcup T \subseteq X$ so $h(A) \subseteq h(\bigcup T)$. Therefore $x \in h(\bigcup T)$ and so $\bigcup T \subseteq h(\bigcup T)$. Let $x \in h(\bigcup T)$ be given. $\bigcup T \subseteq h(\bigcup T) \subseteq X$ so $h(\bigcup T) \subseteq h(h(\bigcup T))$. Therefore $h(\bigcup T) \in T$ and $x \in \bigcup T$. Thus $h(\bigcup T) \subseteq \bigcup T$ and $\bigcup T = h(\bigcup T)$.

Lemma 2

Given sets X and Y and injections $i : X \to Y$ and $j : Y \to X$, let $* : P(X) \to P(X)$ be defined as $A^* = X - j[Y - i[A]]$ for each $A \subseteq X$. Then $A \subseteq B \subseteq X \to A^* \subseteq B^*$.

Let $x \in A^*$ be given. Then $x \notin j[Y - i[A]]$. Suppose $x \notin j[Y]$. Then since $j[Y - i[B]] \subseteq j[Y - i[A]]$, $x \in X - j[Y - i[B]] = B^*$. Otherwise let $y = j^{-1}(x)$. $y \notin Y - i[A]$, so $y \in i[A]$. Therefore let $a = i^{-1}(y)$. Thus $a \in A$ and since $A \subseteq B$, $a \in B$. This means that $y \in i[B]$, $y \notin Y - i[B]$, $x \notin j[Y - i[B]]$, and $x \in X - j[Y - i[B]] = B^*$. Therefore $A^* \subseteq B^*$

Theorem

From the lemmas we get $\exists T \subseteq X$ such that $T = T^*$. Let $f = \begin{cases} i(x) & x \in T \\ j^{-1}(x) & x \notin T \end{cases}$. Then f is a bijection.

Let $x_1 \neq x_2$ be given. If $x_1 \in T \wedge x_2 \in T$ or $x_1 \notin T \wedge x_2 \notin T$ then we are done by the injectiveness of i and j respectively. Therefore let $x_1 \in T$ and $x_2 \notin T$ (the other case is symmetric). $f(x_1) = i(x_1) \in i(T)$. $f(x_2) = j^{-1}(x_2) \notin i(T)$. Therefore $f(x_1) \neq f(x_2)$ and so f is injective. Let $y \in Y$ be given. Suppose $y \in i[T]$. Then $\exists t \in T$ such that $i(t) = y$. Thus $f(t) = i(t) = y$. Otherwise $y \notin i[T]$. This means $y \in Y - i[T]$. Thus $j(y) \in j[Y - i(T)]$ and $j(y) \notin T$. $f(j(y)) = j^{-1}(j(y)) = y$. Therfore f is surjective and so f is bijective

Appendix 2: Automated Proofs of Various Theorems

Note that AProS automatically removes unused premises

Transitivity of Subset

1.	$x \subseteq y$	Prem
2.	$y \subseteq z$	Prem
3.	$u \in x$	Assum
4.	$(\forall z1)(z1 \in y \rightarrow z1 \in z)$	DefE 2
5.	$u \in y \rightarrow u \in z$	∀E 4
6.	$(\forall z)(z \in x \rightarrow z \in y)$	DefE 1
7.	$u \in x \rightarrow u \in y$	∀E 6
8.	$u \in y$	→E 7, 3
9.	$u \in z$	→E 5, 8
10.	$u \in x \rightarrow u \in z$	→I 9
11.	$(\forall z1)(z1 \in x \rightarrow z1 \in z)$	∀I 10
12.	$x \subseteq z$	DefI 11

$\bigcup P(X) = X$

1.	$u \in x$	Assum
2.	$v \in x$	Assum
3.	$v \in x \to v \in x$	→I 2
4.	$(\forall z)(z \in x \to z \in x)$	∀I 3
5.	$x \subseteq x$	DefI 4
6.	$x \in P(x)$	DefI 5
7.	$x \in P(x) \;\&\; u \in x$	&I 6, 1
8.	$(\exists y)(y \in P(x) \;\&\; u \in y)$	∃I 7
9.	$u \in \bigcup P(x)$	DefI 8
10.	$u \in x \to u \in \bigcup P(x)$	→I 9
11.	$(\forall z)(z \in x \to z \in \bigcup P(x))$	∀I 10
12.	$x \subseteq \bigcup P(x)$	DefI 11
13.	$w \in \bigcup P(x)$	Assum
14.	$(\exists y)(y \in P(x) \;\&\; w \in y)$	DefE 13
15.	$u1 \in P(x) \;\&\; w \in u1$	Assum
16.	$u1 \in P(x)$	&E, L 15
17.	$u1 \subseteq x$	DefE 16
18.	$(\forall z)(z \in u1 \to z \in x)$	DefE 17
19.	$w \in u1 \to w \in x$	∀E 18
20.	$w \in u1$	&E, R 15
21.	$w \in x$	→E 19, 20
22.	$w \in x$	∃E 14, 21
23.	$w \in \bigcup P(x) \to w \in x$	→I 22
24.	$(\forall z)(z \in \bigcup P(x) \to z \in x)$	∀I 23
25.	$\bigcup P(x) \subseteq x$	DefI 14
26.	$x \subseteq \bigcup P(x) \;\&\; \bigcup P(x) \subseteq x$	&I 12, 25
27.	$x = \bigcup P(x)$	DefI 26

Lemma 1

1. $(\forall x1)(\forall x2)(x1 \subseteq x2 \to h(x1) \subseteq h(x2))$ Prem
2. $u \in \bigcup t$ Assum
3. $(\exists y)(y \in t \,\&\, u \in y)$ DefE 2
4. $v \in t \,\&\, u \in v$ Assum
5. $(\forall x2)(v \subseteq x2 \to h(v) \subseteq h(x2))$ ∀E 1
6. $v \subseteq \bigcup t \to h(v) \subseteq h(\bigcup t)$ ∀E 5
7. $w \in v$ Assum
8. $v \in t$ &E, L 4
9. $v \in t \,\&\, w \in v$ &I 8, 7
10. $(\exists y)(y \in t \,\&\, w \in y)$ ∃I 9
11. $w \in \bigcup t$ DefI 10
12. $w \in v \to w \in \bigcup t$ →I 11
13. $(\forall z)(z \in v \to z \in \bigcup t)$ ∀I 12
14. $v \subseteq \bigcup t$ DefI 13
15. $h(v) \subseteq h(\bigcup t)$ →E 6, 14
16. $(\forall z)(z \in h(v) \subseteq z \in h(\bigcup t))$ DefE 15
17. $u \in h(v) \to u \in h(\bigcup t)$ ∀E 16
18. $v \in t$ &E, L 4
19. $v \subseteq h(v)$ DefE 18
20. $(\forall z)(z \in v \to z \in h(v))$ DefE 19
21. $u \in v \to u \in h(v)$ ∀E 20
22. $u \in v$ &E, R 4
23. $u \in h(v)$ →E 21, 22
24. $u \in h(\bigcup t)$ →E 17, 23
25. $u \in h(\bigcup t)$ ∃E 3, 24
26. $u \in \bigcup t \to u \in h(\bigcup t)$ →I 25
27. $(\forall z)(z \in \bigcup t \to z \in h(\bigcup t))$ ∀I 26
28. $\bigcup t \subseteq h(\bigcup t)$ DefI 27
29. $x \in h(\bigcup t)$ Assum
30. $(\forall x2)(\bigcup(t) \subseteq x2 \to h(\bigcup t) \subseteq h(x2))$ ∀E 1
31. $\bigcup t \subseteq h(\bigcup t) \to h(\bigcup t) \subseteq h(h(\bigcup t))$ ∀E 30
32. $u1 \in \bigcup t$ Assum

33.	$(\exists y)(y \in t \,\&\, u1 \in y)$	DefE 32
34.	$v1 \in t \,\&\, u1 \in v1$	Assum
35.	$(\forall x2)(v1 \subseteq x2 \to h(v1) \subseteq h(x2))$	∀E 1
36.	$v1 \subseteq \bigcup t \to h(v1) \subseteq h(\bigcup t)$	∀E 35
37.	$w1 \in v1$	Assum
38.	$v1 \in t$	&E, L 34
39.	$v1 \in t \,\&\, w1 \in v1$	&I 38, 37
40.	$(\exists y)(y \in t \,\&\, w1 \in y)$	∃I 39
41.	$w1 \in \bigcup t$	DefI 40
42.	$w1 \in v1 \to w1 \in \bigcup t$	→I 41
43.	$(\forall z)(z \in v1 \to z \in \bigcup t)$	∀I 42
44.	$v1 \subseteq \bigcup t$	DefI 43
45.	$h(v1) \subseteq h(\bigcup t)$	→E 36, 44
46.	$(\forall z)(z \in h(v1) \subseteq z \in h(\bigcup t))$	DefE 45
47.	$u1 \in h(v1) \to u1 \in h(\bigcup t)$	∀E 46
48.	$v1 \in t$	&E, L 34
49.	$v1 \subseteq h(v1)$	DefE 48
50.	$(\forall z)(z \in v1 \to z \in h(v1))$	DefE 49
51.	$u1 \in v1 \to u1 \in h(v1)$	∀E 50
52.	$u1 \in v1$	&E, R 34
53.	$u1 \in h(v1)$	→E 51, 52
54.	$u1 \in h(\bigcup t)$	→E 47, 53
55.	$u1 \in h(\bigcup t)$	∃E 33, 54
56.	$u1 \in \bigcup t \to u1 \in h(\bigcup t)$	→I 55
57.	$(\forall z)(z \in \bigcup t \to z \in h(\bigcup t))$	∀I 56
58.	$\bigcup t \subseteq h(\bigcup t)$	DefI 57
59.	$h(\bigcup t) \subseteq h(h(\bigcup t))$	→E 31,58
60.	$h(\bigcup t) \in t$	DefE 59
61.	$h(\bigcup t) \in t \,\&\, x \in h(\bigcup t)$	&I 60, 29
62.	$(\exists y)(y \in t \,\&\, x \in y)$	∃I 61
63.	$x \in \bigcup t$	DefI 62
64.	$x \in h(\bigcup t) \to x \in \bigcup t$	→I 63
65.	$(\forall z)(z \in h(\bigcup t) \to z \in \bigcup t)$	∀I 64

66. $h(\bigcup t) \subseteq \bigcup t$ DefI 65

67. $\bigcup t \subseteq h(\bigcup t)\ \&\ h(\bigcup t) \subseteq \bigcup t$ &I 28, 66

68. $\bigcup t = h(\bigcup t)$ DefI 67

Lemma 2

1. $x \subseteq y$ Prem
2. $u \in x^*$ Assum
3. $u \in a - j[b - i[x]]$ DefE 2
4. $u \in a \,\&\, \sim u \in j[b - i[x]]$ DefE 3
5. $u \in a$ &E, L 4
6. $u \in j[b - i[y]]$ Assum
7. $(\exists z)(z \in b - i[y] \,\&\, j(z) = u)$ DefE 6
8. $v \in b - i[y] \,\&\, j(v) = u$ Assum
9. $v \in b - i[y]$ &E, L 8
10. $v \in b \,\&\, \sim v \in i[y]$ DefE 9
11. $v \in b$ &E, L 10
12. $v \in i[x]$ Assum
13. $(\exists y)(y \in x \,\&\, i(y) = v)$ DefE 12
14. $w \in x \,\&\, i(w) = v$ Assum
15. $v \in b - i[y]$ &E, L 8
16. $v \in b \,\&\, \sim v \in i[y]$ DefE 15
17. $v \in b$ &E, L 16
18. $v \in i[x]$ Assum
19. $(\forall z)(z \in x \rightarrow z \in y)$ DefE 1
20. $w \in x \rightarrow w \in y$ ∀E 19
21. $w \in x$ &E, L 14
22. $w \in y$ →E 20,21
23. $i(w) = v$ &E, R 14
24. $(w \in y \,\&\, i(w) = v$ &I 22,23
25. $(\exists z)(z \in y \,\&\, i(z) = v)$ ∃I 24
26. $v \in i[y]$ DefI 25
27. $v \in b - i[y]$ &E, L 8
28. $v \in b \,\&\, \sim v \in i[y]$ DefE 27
29. $\sim v \in i[y]$ &E, R 28
30. $\perp$ ⊥I 26,29
31. $\sim v \in i[x]$ ∼I 30
32. $v \in b \,\&\, \sim v \in i[x]$ &I 17,31

33. $v \in b - i[x]$ DefI 32

34. $j(v) = u$ &E, R 8

35. $v \in b - i[x] \,\&\, j(v) = u$ &I 33,34

36. $(\exists y)(y \in b - i[x] \,\&\, j(y) = u)$ ∃I 35

37. $u \in j[b - i[x]]$ DefI 36

38. $u \in j[b - i[x]]$ ∃E 13,37

39. $u \in a - j[b - i[x]]$ DefE 2

40. $u \in a \,\&\, \sim u \in j[b - i[x]]$ DefE 39

41. $\sim u \in j[b - i[x]]$ &E, R 40

42. $\perp$ ⊥I 38,41

43. $\sim v \in i[x]$ ∼I 42

44. $v \in b \,\&\, \sim v \in i[x]$ &I 11,43

45. $v \in b - i[x]$ DefI 44

46. $j(v) = u$ &E, R 8

47. $v \in b - i[x] \,\&\, j(v) = u$ &I 45,46

48. $(\exists y)(y \in b - i[x] \,\&\, j(y) = u)$ ∃I 47

49. $u \in j[b - i[x]$ DefE 48

50. $u \in a - j[b - i[x]]$ DefE 2

51. $u \in a \,\&\, \sim u \in j[b - i[x]]$ DefE 50

52. $\sim u \in j[b - i[x]]$ &E, R 51

53. $\perp$ ⊥I 49, 52

54. $\perp$ ∃E 7, 53

55. $\sim u \in j[b - i[y]]$ ∼I 54

56. $u \in a \,\&\, \sim u \in j[b - i[y]]$ &I 5, 55

57. $u \in a - j[b - i[y]]$ DefI 56

58. $u \in y^*$ DefI 57

59. $u \in x^* \rightarrow u \in y^*$ →I 58

60. $(\forall z)(z \in x^* \rightarrow u \in y^*)$ ∀I 59

61. $x^* \subseteq y^*$ DefI 60

The Identity Function is Injective

1.	$\sim v = u$	Assum
2.	$(v, w) \in i \,\&\, (u, w) \in i$	Assum
3.	$u1 \in v$	Assum
4.	$(u, w) \in i$	&E, R 2
5.	$u = w$	DefE 4
6.	$u \subseteq w \,\&\, w \subseteq u$	DefE 5
7.	$w \subseteq u$	&E, R 6
8.	$(\forall z)(z \in w \to z \in u)$	DefE 7
9.	$u1 \in w \to u1 \in u$	∀E 8
10.	$(v, w) \in i$	&E, L 2
11.	$v = w$	DefE 10
12.	$v \subseteq w \,\&\, w \subseteq v$	DefE 11
13.	$v \subseteq w$	&E, L 12
14.	$(\forall z)(z \in v \to z \in w)$	DefE 13
15.	$u1 \in v \to u1 \in w$	∀E 14
16.	$u1 \in w$	→E 15, 3
17.	$u1 \in u$	→E 9, 16
18.	$u1 \in v \to u1 \in u$	→I 17
19.	$(\forall z)(z \in v \to z \in u)$	∀I 18
20.	$v \subseteq u$	DefI 19
21.	$v1 \in u$	Assum
22.	$(v, w) \in i$	&E, L 2
23.	$v = w$	DefE 22
24.	$v \subseteq w \,\&\, w \subseteq v$	DefE 23
25.	$w \subseteq v$	&E, R 24
26.	$(\forall z)(z \in w \to z \in v)$	DefE 25
27.	$v1 \in w \to v1 \in v$	∀E 26
28.	$(u, w) \in i$	&E, R 2
29.	$u = w$	DefE 29
30.	$u \subseteq w \,\&\, w \subseteq u$	DefE 29
31.	$u \subseteq w$	&E, L 30
32.	$(\forall z)(z \in u \to z \in w)$	DefE 31

33.	$v1 \in u \rightarrow v1 \in w$	∀E 32
34.	$v1 \in w$	→E 33, 21
35.	$v1 \in v$	→E 27, 34
36.	$v1 \in u \rightarrow v1 \in v$	→I 35
37.	$(\forall z)(z \in u \rightarrow z \in v)$	∀I 36
38.	$u \subseteq v$	DefI 37
39.	$v \subseteq u \,\&\, u \subseteq v$	&I 20,38
40.	$v = u$	DefI 39
41.	$\perp$	⊥I 40,1
42.	$(\sim ((v, w) \in i \,\&\, (u, w) \in i))$	∼I 41
43.	$(\forall z)(\sim ((v, z) \in i \,\&\, (u, z) \in i))$	∀I 42
44.	$(\sim v = u \rightarrow (\forall z)(\sim ((v, z) \in i \,\&\, (u, z) \in i))$	→I 43
45.	$(\forall y)(\sim v = y \rightarrow (\forall z)(\sim ((v, z) \in i \,\&\, (y, z) \in i))$	∀I 44
46.	$(\forall x)(\forall y)(\sim x = y \rightarrow (\forall z)(\sim ((x, z) \in i \,\&\, (y, z) \in i))$	∀I 45
47.	$I(i)$	DefI 46

Appendix 3: A Partially Automated Proof of the Theorem

Note that AProS automatically removes unused premises

Setting up the four cases

1. $u \in a \,\&\, v \in a$ Assum
2. $f(u) = f(v)$ Assum
3. $(i(u) = f(v) \,\&\, u \in t) \vee (j(f(v)) = u \,\&\, \sim u \in t)$ DefE 2
4. $i(u) = f(v) \,\&\, u \in t$ Assum
5. $i(u) = f(v)$ &E, L 4
6. $f(v) = i(u)$ DefE 5
7. $v \in a \,\&\, i(u) \in b \,\&\, ((i(v) = i(u) \,\&\, v \in t) \vee (j(i(u)) = v \,\&\, \sim v \in t))$ DefE 6
8. $(i(v) = i(u) \,\&\, v \in t) \vee (j(i(u)) = v \,\&\, \sim v \in t)$ &E, R 7
9. $i(v) = i(u) \,\&\, v \in t$ Assum
10. $u = v$ Case 1
11. $j(i(u)) = v \,\&\, \sim v \in t$ Assum
12. $u = v$ Case 2
13. $u = v$ ∨E 8, 10, 12
14. $j(f(v)) = u \,\&\, \sim u \in t$ Assum
15. $j(f(v)) = u$ &E, L 14
16. $f(v) = j^{-1}(u)$ DefE 15
17. $v \in a \,\&\, j^{-1}(u) \in b \,\&\, ((i(v) = j^{-1}(u) \,\&\, v \in t) \vee (j(j^{-1}(u)) = v \,\&\, \sim v \in t))$ DefE 16
18. $(i(v) = j^{-1}(u) \,\&\, v \in t) \vee (j(j^{-1}(u)) = v \,\&\, \sim v \in t)$ &E, R 17
19. $i(v) = j^{-1}(u) \,\&\, v \in t$ Assum
20. $u = v$ Case 3
21. $j(j^{-1}(u)) = v \,\&\, \sim v \in t$ Assum
22. $u = v$ Case 4
23. $u = v$ ∨E 18,20,22
24. $u = v$ ∨E 3, 13, 23
25. $f(u) = f(v) \rightarrow u = v$ →I 24
26. $(u \in a \,\&\, v \in a) \rightarrow (f(u) = f(v) \rightarrow u = v)$ →I 25
27. $(\forall y)((u \in a \,\&\, y \in a) \rightarrow (f(u) = f(y) \rightarrow u = y))$ ∀I 26
28. $(\forall x)(\forall y)((x \in a \,\&\, y \in a) \rightarrow (f(x) = f(y) \rightarrow x = y))$ ∀I 27
29. $I(f, a, b)$ DefI 28

Case 1

1. $I(i, a, b)$ Prem
2. $u \in a \,\&\, v \in a$ Prem
3. $i(v) = i(u)$ Prem
4. $F(i, a, b) \,\&\, (\forall x)(\forall y)((x \in a \,\&\, y \in a) \to (i(x) = i(y) \to x = y))$ DefE 1
5. $(\forall x)(\forall y)((x \in a \,\&\, y \in a) \to (i(x) = i(y) \to x = y))$ &E, R 4
6. $(\forall y)((u \in a \,\&\, y \in a) \to (i(u) = i(y) \to u = y))$ ∀E 5
7. $(u \in a \,\&\, v \in a) \to (i(u) = i(v) \to u = v)$ ∀E 6
8. $i(u) = i(v) \to u = v$ →E 7, 2
9. $i(u) = i(v)$ DefE 3
10. $u = v$ →E 8,9

Case 4

1. $j(j^{-1}(u)) = v$ Prem
2. $(\forall x)(\forall y)(j^{-1}(x) = j^{-1}(y) \to x = y)$ Prem
3. $(\forall y)(j^{-1}(u) = j^{-1}(y) \to u = y)$ ∀E 2
4. $j^{-1}(u) = j^{-1}(v) \to u = v$ ∀E 3
5. $j^{-1}(u) = j^{-1}(v)$ DefE 1
6. $u = v$ →E 4, 5

The Setup of Case 3 by Hand (and by Symmetry Case 2)

1.	$t = t^*$	Prem
2.	$u \in a \,\&\, v \in a$	Prem
3.	$j(f(v)) = u \,\&\, \sim u \in t$	Prem
4.	$t \subseteq t^* \,\&\, t^* \subseteq t$	DefE 1
5.	$t^* \subseteq t$	&E, R 4
6.	$(\forall z)(z \in t^* \rightarrow z \in t)$	DefE 5
7.	$u \in t^* \rightarrow u \in t$	∀E 6
8.	$u \in a$	&E, L 2
9.	$u \in j[b - i[t]]$	Assum
10.	$(\exists y)(y \in b - i[t] \,\&\, j(y) = u)$	DefE 9
11.	$z \in b - i[t] \,\&\, j(z) = u$	Assum
12.	$z \in i[t]$	From Rest
13.	$\sim z \in i[t]$	From Rest
14.	$\bot$	⊥I 12, 13
15.	$\bot$	∃E 10, 14
16.	$\sim u \in j[b - i[t]]$	∼I 15
17.	$u \in a \,\&\, \sim u \in j[b - i[t]]$	&I 8, 16
18.	$u \in a - j[b - i[t]]$	DefI 17
19.	$u \in t^*$	DefI 18
20.	$u \in t$	→E 7, 19
21.	$\sim u \in t$	&E, R 3
22.	$\bot$	⊥I 20, 21
23.	$u = v$	∼Q 22

The Rest of Case 3

1.	$i(v) = j^{-1}(u) \,\&\, v \in t$	Prem
2.	$z \in b - i[t] \,\&\, j(z) = u$	Prem
3.	$v \in t$	&E, R 1
4.	$u1 \in i(v)$	Assum
5.	$j(z) = u$	&E, R 2
6.	$z = j^{-1}(u)$	DefE 5
7.	$z \subseteq j^{-1}(u) \,\&\, j^{-1}(u) \subseteq z$	DefE 6
8.	$j^{-1}(u) \subseteq z$	&E, R 7
9.	$(\forall z1)(z1 \in j^{-1}(u) \rightarrow z1 \in z)$	DefE 8
10.	$u1 \in j^{-1}(u) \rightarrow u1 \in z$	∀E 9
11.	$i(v) = j^{-1}(u)$	&E, L 1
12.	$i(v) \subseteq j^{-1}(u) \,\&\, j^{-1}(u) \subseteq i(v)$	DefE 11
13.	$i(v) \subseteq j^{-1}(u)$	&E, L 12
14.	$(\forall z)(z \in i(v) \rightarrow z \in j^{-1}(u))$	DefE 13
15.	$u1 \in i(v) \rightarrow u1 \in j^{-1}(u)$	∀E 14
16.	$u1 \in j^{-1}(u)$	→E 15, 4
17.	$u1 \in z$	→E 10, 16
18.	$u1 \in i(v) \rightarrow u1 \in z$	→I 17
19.	$(\forall z1)(z1 \in i(v) \rightarrow z1 \in z)$	∀I 18
20.	$i(v) \subseteq z$	DefI 19
21.	$v1 \in z$	Assum
22.	$i(v) = j^{-1}(u)$	&E, L 1
23.	$i(v) \subseteq j^{-1}(u) \,\&\, j^{-1}(u) \subseteq i(v)$	DefE 22
24.	$j^{-1}(u) \subseteq i(v)$	&E, R 23
25.	$(\forall z)(z \in j^{-1}(u) \rightarrow z \in i(v))$	DefE 24
26.	$v1 \in j^{-1}(u) \rightarrow v1 \in i(v)$	∀E 25
27.	$j(z) = u$	&E, R 2
28.	$z = j^{-1}(u)$	DefE 27
29.	$z \subseteq j^{-1}(u) \,\&\, j^{-1}(u) \subseteq z$	DefE 28
30.	$z \subseteq j^{-1}(u)$	&E, L 29
31.	$(\forall z1)(z1 \in z \rightarrow z1 \in j^{-1}(u))$	DefE 30
32.	$v1 \in z \rightarrow v1 \in j^{-1}(u)$	∀E 31

33. $v1 \in j^{-1}(u)$ →E 32, 21

34. $v1 \in i(v)$ →E 26, 33

35. $v1 \in z \rightarrow v1 \in i(v)$ →I 34

36. $(\forall z1)(z1 \in z \rightarrow z1 \in i(v))$ ∀I 35

37. $z \subseteq i(v)$ DefI 36

38. $i(v) \subseteq z \,\&\, z \subseteq i(v)$ &I 20, 37

39. $i(v) = z$ DefI 38

40. $v \in t \,\&\, i(v) = z$ &I 3, 39

41. $(\exists y)(y \in t \,\&\, i(y) = z)$ ∃I 40

42. $z \in i[t]$ DefI 41

43. $z \in b - i[t]$ &E, L 2

44. $z \in b \,\&\, \sim z \in i[t]$ DefE 43

45. $\sim z \in i[t]$ &E, R 44

46. $z \in i[t] \,\&\, \sim z \in i[t]$ &I 42, 45