Discovering Theorems using GOEDEL: a case study
Calculemus’2001, Siena, Italy, 21 June 2001

Johan Gijsbertus Frederik Belinfante
Georgia Institute of Technology, Atlanta, GA 30332–0160 (U.S.A.)
belinfan@math.gatech.edu

Abstract.

The GOEDEL program is a Mathematica™ implementation of Gödel’s algorithm for class formation. It is mainly used to formulate definitions needed for Otter proofs in set theory. The NBG axioms for classes are used, as modified by Quaife.

The GOEDEL program can also discover new facts. For example, we discovered:

\[ \text{image[inverse[TC],P[x]]} = P[H[x]] \]

and other facts about the hereditary core \( H[x] \) of a class \( x \), which were subsequently proved using Otter.

A recent version of the GOEDEL program, and complete details of all Otter proofs can be found on the author’s website:

http://www.math.gatech.edu/~belinfan/research/
Notation

\[ a \rightarrow b \text{ means Mathematica input } a \text{ gives output } b. \]

Gödel’s 9 primitives

1. universal class: \( \{x \mid \text{True} \} \rightarrow V \)

2. membership relation: \( \{\langle u, v \rangle \mid u \in v \} \rightarrow E \)

3. complement: \( \{u \mid \text{not}[u \in x]\} \rightarrow \text{complement}[x] \)

4. domain: \( \{u \mid \exists v \langle u, v \rangle \in x\} \rightarrow \text{domain}[x] \)

5. flip: \( \{\langle\langle u, v \rangle, w \rangle \mid \langle v, u \rangle, w \rangle \in x\} \rightarrow \text{flip}[x] \)

6. rotate: \( \{\langle\langle u, v \rangle, w \rangle \mid \langle v, w \rangle, u \rangle \in x\} \rightarrow \text{rotate}[x] \)

7. pairset (unordered pair):

\[ \text{assert}[\{w \mid (w = x) \text{ or } (w = y)\} \rightarrow \text{assert}[\{x, y\}] \rightarrow \text{True} \]

8. cartesian product: \( \{\langle u, v \rangle \mid u \in x \land v \in y\} \rightarrow x \times y \)

9. intersection: \( \{u \mid u \in x \land u \in y\} \rightarrow x \cap y \)
Defined Concepts

inverse: \( \{ \langle u, v \rangle \mid \langle v, u \rangle \in x \} \rightarrow \text{inverse}[x] \)
\[ \text{domain[flip}[x \times V]\text{]} \rightarrow \text{inverse}[x] \]

quantifiers in the GOEDEL program are restricted to sets

range: \( \{ v \mid \exists u \langle u, v \rangle \in x \} \rightarrow \text{range}[x] \)
\[ \text{domain[\text{inverse}[x]]} \rightarrow \text{range}[x] \]

image: \( \{ v \mid \exists u \langle u, v \rangle \in x \land u \in y \} \rightarrow \text{image}[x, y] \)
\[ \text{range}[x \cap (y \times V)] \rightarrow \text{image}[x, y] \]

composite: \( \{ \langle u, v \rangle \mid \exists w \langle v, w \rangle \in x \land \langle u, w \rangle \in y \} \rightarrow x \circ y \)
\[ \text{domain[rotate[flip}[x \times V]\text{]} \cap \text{flip[rotate}[y \times V]\text{]}]} \rightarrow x \circ y \]

subset relation: \( \{ \langle u, v \rangle \mid u \subset v \} \rightarrow S \)
\[ (V \times V) \cap \text{complement[complement[E]} \circ \text{inverse[E]}\text{]} \rightarrow S \]
Relation between **GOEDEL** and **Otter**

**Mathematica™ program**

**GOEDEL**
Gödel’s algorithm and simplifier

defining classes, formulating theorems & discovering lemmas

verifying rewrite rules

**McCune’s program**

**Otter**
for automated reasoning
Sum and Power Classes

sum class: \( \{ u \mid \exists v \ (u \in v \text{ and } v \in x) \} \rightarrow U[x] \)

image[\text{inverse}[E], x] \rightarrow U[x]

power class: \( \{ u \mid u \subset x \} \rightarrow P[x] \)

\text{complement}[\text{image}[E, \text{complement}[x]]] \rightarrow P[x]

These are related by: \( U[P[x]] \rightarrow x \)

The class FULL

various descriptions of the class of all full sets:

\( \{ x \mid \forall u, v \ ((u \in v \& v \in x) \Rightarrow u \in x) \} \rightarrow \text{FULL} \)

\( \{ x \mid \forall v \ (v \in x \Rightarrow v \subset x) \} \rightarrow \text{FULL} \)

\( \{ x \mid U[x] \subset x \} \rightarrow \text{FULL} \)

definition of \text{FULL} in terms of primitives:

\( \text{complement}[\text{range}[E \cap \text{complement}[S]]] \rightarrow \text{FULL} \)
The functions BIGCUP and POWER

\[ \lambda[x, u[x]] \rightarrow \text{BIGCUP} \quad \lambda[x, p[x]] \rightarrow \text{POWER} \]

\(\text{BIGCUP} \circ \text{POWER} \rightarrow \text{Id} \quad \text{where} \quad \lambda[x, x] \rightarrow \text{Id} \)

**VERTSECT and IMAGE**

\[ \lambda[y, \text{image}[x, \{y\}]] \rightarrow \text{VERTSECT}[x] \]

\[ \lambda[y, \text{image}[x, y]] \rightarrow \text{IMAGE}[x] \]

\[ \text{IMAGE}[x] \circ \text{SINGLETON} \rightarrow \text{VERTSECT}[x] \]

\[ \text{where} \quad \lambda[x, \{x\}] \rightarrow \text{SINGLETON} \]

\[ \text{VERTSECT}[x \circ \text{inverse}[E]] \rightarrow \text{IMAGE}[x] \]

**Some applications**

\[ \text{IMAGE}[\text{inverse}[E]] \rightarrow \text{BIGCUP} \]

\[ \text{VERTSECT}[\text{inverse}[S]] \rightarrow \text{POWER} \]

\[ \lambda[y, x \cap y] \rightarrow \text{IMAGE}[\text{id}[x]] \quad \text{where: Id} \cap (x \times x) \rightarrow \text{id}[x] \]

Do not confuse this with \(\lambda[\langle x, y \rangle, x \cap y] \rightarrow \text{CAP} \)

**Defining \(\lambda\) itself**

\[ \lambda[x, e] := \text{Module}[\{y = \text{Unique}[\ ], \}
\]

\[ \text{VERTSECT}[\{\langle x, y \rangle \mid y \in e\} \circ \text{id}\{x \mid \text{True}\}] \]
Transitive Closure $tc[x]$

The transitive closure $tc[x]$ is the smallest full class containing $x$.

$$\lambda[x, A[\text{FULL} \cap \text{image}[S, \{x\}]]] \rightarrow TC,$$

where: \( \{u \mid \forall v (v \in x \Rightarrow u \in v)\} \rightarrow A[x] \)

and \( \{u \mid x \subset u\} \rightarrow \text{image}[S, \{x\}] \).

The domain of $TC$ is $U[\text{FULL}] = V$.

The function $TC$ is idempotent: $TC \circ TC \rightarrow TC$.

$U[\text{image}[TC, P[x]]] \rightarrow tc[x]$ holds for any class $x$.

$\lambda[x, tc[x]] \rightarrow TC$.

Hereditary Core $H[x]$ 

$H[x]$ is the largest full subclass of $x$.

Definition: $U[\text{FULL} \cap P[x]] \rightarrow H[x]$.

The proofs about $H[x]$ depend on facts about $tc[x]$.

$\lambda[x, H[x]] \rightarrow HC$ is idempotent: $HC \circ HC \rightarrow HC$.

Definition: $\text{BIGCUP} \circ \text{IMAGE}[\text{id}[\text{FULL}]] \circ \text{POWER} \rightarrow HC$.

$U[\text{image}[HC, P[x]]] \rightarrow H[x]$ holds for any class $x$.

The class $\Omega$ of ordinals: $AxReg \Rightarrow H[\text{FULL}] = \Omega$. 
VSNormality

VSNormality is one of several tools which are useful for discovering formulas for a specified relation.

\[
\text{VSNormality}[x_] := \text{Module}[\{u = \text{Unique}[1], v = \text{Unique}[1]\}, \\
(u \in V) = \text{True}; (v \in V) = \text{True}; \\
\text{Id} \circ x == \{\langle u, v \rangle | v \in \text{image}[x, \{u\}]\}]
\]

**A few discoveries**

Here are facts discovered using **GOEDEL**, subsequently proved using **Otter**, and now incorporated in the **GOEDEL** program.

1. \(\text{inverse}[\text{HC}] \circ S \rightarrow S \circ \text{TC}\)
2. \(\text{image}[\text{inverse}[\text{TC}], P[x]] \rightarrow P[H[x]]\)

Lesser discoveries, not added to the **GOEDEL** program:

3. \(\text{BIGCUP} \circ \text{TC} \circ \text{SINGLETON} // \text{VSNormality} \rightarrow\)
   \(\text{BIGCUP} \circ \text{TC} \circ \text{SINGLETON} == \text{TC}\).
4. \(\text{complement}[\text{HC}] \circ \text{HC} // \text{VSNormality} \rightarrow\)
   \(\text{complement}[\text{HC}] \circ \text{HC} == \text{Id} \circ \text{complement}[\text{HC}]\)
Conclusions

1. Automated reasoning in mathematics must be able to cope with sets. By working in NBG set theory, a conservative extension of ZF set theory that can be finitely axiomatized, one can take advantage of Gödel’s algorithm to work within first-order logic.

2. The GOEDEL program is mainly used as a preprocessing tool to help formulate definitions, simplify descriptions of classes and statements about classes, and to discover lemmas. Tools like VSNormality are available for simplifying specific classes.

3. Gödel’s algorithm can be used to eliminate all quantifiers over sets, any statement in set theory can be formulated as an equation, which allows one to work at the term level.

4. The interactive nature of the GOEDEL program lends itself well to experimentation and discovery. The synergistic interplay between Otter and the GOEDEL program provides an environment that is conducive to making progress in applying automated reasoning to set theory.