Reasoning about Iteration in Gödel’s Class Theory
CADE 19, Miami, Florida, 2003 July 31

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

Abstract.
The NBG axioms for classes are used, as modified by Quaife.
The focus of the talk is on how the author’s GOEDEL program was used to formulate the definition of the class constructor iterate[x,y], to discover its basic properties, and how to use it for developing arithmetic, and other applications of interest.

A key idea is to avoid explicit induction by using instead the uniqueness theorem for iteration:

\[
\text{image}[w, \{0\}] = y \land w \circ \text{SUCC} = x \circ w \\
\Rightarrow w \circ \text{id}[\omega] = \text{iterate}[x,y].
\]

Complete details of proofs can be found on the author’s website:
http://www.math.gatech.edu/~belinfan/research/
Relation between GOEDEL and Otter

Mathematica\textsuperscript{TM} program

GOEDEL
Gödel’s algorithm and simplifier

defining classes, formulating theorems & discovering lemmas

verifying rewrite rules

McCune’s program

Otter
for automated reasoning
Basic Concepts

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

set of natural numbers: \( \omega = \{0, 1, 2, \ldots \}. \)

successor: \( \text{succ}[x] = x \cup \{x\}. \)

successor function: \( \text{SUCC} = \lambda x. \text{succ}[x]. \)

induction: \( 0 \in x \land \text{image}[\text{SUCC}, x] \subset x \Rightarrow \omega \subset x. \)

quantifiers in the GOEDEL program are restricted to sets

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

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

image: \( \{ v \mid \exists u \langle u, v \rangle \in x \land u \in y \} = \text{image}[x, y] \)

fixpoint class: \( \{ v \mid \langle v, v \rangle \in x \} = \text{fix}[x] \)

identity on \( x \): \( \{ \langle u, v \rangle \mid u = v \land u \in x \} = \text{id}[x] \)

global identity: \( \text{id}[V] = \text{Id} \)

These are related as follows:

\( \text{id}[x] \subset y \iff x \subset \text{fix}[y] \)

\( \text{range}[x \circ \text{id}[y]] = \text{image}[x, y] \)
Flip and Rotate

The functions \texttt{FIRST} and \texttt{SECOND} project out the components of an ordered pair, while \texttt{SWAP} interchanges them:

\[
\text{FIRST} = \{ \langle \langle u, v \rangle, w \rangle | w = u \} \\
\text{SECOND} = \{ \langle \langle u, v \rangle, w \rangle | w = v \} \\
\text{SWAP} = \{ \langle \langle u, v \rangle, w \rangle | w = \langle v, u \rangle \}
\]

Gödel's primitives \texttt{flip}[x] and \texttt{rotate}[x] could be expressed in terms of these functions:

\[
\text{flip}[x] = \{ \langle \langle u, v \rangle, w \rangle | \langle \langle v, u \rangle, w \rangle \in x \} = x \circ \text{SWAP}
\]

\[
\text{rotate}[x] = \{ \langle \langle u, v \rangle, w \rangle | \langle \langle v, w \rangle, u \rangle \in x \} = \text{SECOND} \circ (\text{inverse}[\text{FIRST}] \circ \text{SECOND}) \cap (\text{inverse}[x] \circ \text{FIRST})
\]

One can use these formulas to eliminate \texttt{flip}. One could also eliminate \texttt{rotate}, but there are excellent reasons not to do so.

1. Both \texttt{flip} and \texttt{rotate} preserve unions, intersections and relative complements, whereas composition only preserves unions.

2. Rotation is useful in arithmetic: subtraction and division are related to addition and multiplication by rotation.

3. Rotation formulas are convenient for deriving properties of many other binary functions.
Subvariance

Cross product: \( x \otimes y = \{ \langle \langle s, t \rangle, \langle u, v \rangle \rangle \mid \langle s, u \rangle \in x \land \langle t, v \rangle \in y \} \)

\[
\text{image}[x \otimes y, z] = y \circ z \circ \text{inverse}[x]
\]

Definition: \( \text{subvariant}[x, y] \iff y \subseteq \text{image}[x, y] \)

\[
\text{subvar}[x] = \{ y \mid \text{subvariant}[x, y] \}
\]

Building in initial conditions: the idea

\[
x \subseteq y \cup \text{image}[z, x] \iff \text{subvariant}[\text{id}[y] \cup z, x]
\]

Upper bound relation associated with a relation \( x \):

\[
\text{UB}[x] = \{ \langle u, v \rangle \mid \forall w (w \in u \Rightarrow \langle w, v \rangle \in x) \}.
\]

Quaife’s definition of the subset relation simplifies to \( S = \text{UB}[E] \).

where \( E \) is the membership relation: \( E = \{ \langle u, v \rangle \mid u \in v \} \)

Definition: \( \text{GREATEST}[x] = \text{UB}[x] \cap \text{inverse}[E] \).

Connection between \text{subvar} and \text{GREATEST}:

\[
\text{complement}[	ext{subvar}[x]] = \text{domain}[	ext{GREATEST}[	ext{complement}[x]]]
\]
iterate, power and applications

Definition: \( \text{iterate}[x, y] = \)
\[ \cup [\text{subvar}(((\text{SUCC} \circ \text{id}[\omega]) \otimes x) \cup \text{id}[\{0\} \times y])] \]

Definition: \( \text{power}[x] = \text{iterate}[\text{Id} \otimes x, \text{Id}] \)

Interpretation: \( \text{image}[\text{power}[x], y] \) is the union of all powers of \( x \) whose exponents belong to \( y \).

\[
\text{iterate}[x, y] = \{ \langle u, v \rangle \mid v \in \text{image}[\text{image}[\text{power}[x], \{u\}], y]\}
= \text{SECOND} \circ \text{id}[y \times V] \circ \text{power}[x]
\]

Addition of natural numbers:
\[
\text{NATADD} = \text{id}[\omega] \circ \text{rotate}\text{inverse}[\text{power}[\text{SUCC}]].
\]

Transitive closure of a relation:
\[
\text{trv}[x] = \text{image}[\text{power}[x], \text{complement}[\{0\}]][0]
= (\text{Id} \circ x) \cup (x \circ x) \cup (x \circ x \circ x) \cup \ldots
\]
Addition of Natural Numbers

The key to deriving the laws of addition for natural numbers is the additive law of exponents:

\[
\text{image}[\text{power}[x], y] \circ \text{image}[\text{power}[x], z] = \\
\text{image}[\text{power}[x], \text{image}[\text{NATADD}, \text{cart}[y, z]]].
\]

The uniqueness theorem for \textit{iterate} is used to prove various properties of \texttt{power}[x] from which this and other facts about addition can be derived.

For example, the commutativity of addition follows from the fact that \texttt{image}[\texttt{power}[x], y] commutes with \texttt{image}[\texttt{power}[x], z].

Associativity of addition follows from the associative law for composites of images of \texttt{power[SUCCE]}, together with the fact:

\[
\text{image}[\text{image}[\text{power[SUCCE]}, x], \{0\}] = \omega \cap x.
\]
Subtraction and Multiplication

Subtraction is related to addition via rotation. One shows that \( \text{rotate}[	ext{NATADD}] \) is a function by deriving a cancellation law for addition: \( \text{plus}[x] = \text{NATADD} \circ \text{RIGHT}[x] \) is one-to-one for all \( x \), where \( \text{RIGHT}[x] = \{ \langle u, \langle v, w \rangle \rangle \mid u = v & w = x \} \).

A function \( \text{NATMUL} \) corresponding to multiplication of natural numbers can be defined with the key property
\[
x \in \omega \Rightarrow \text{NATMUL} \circ \text{LEFT}[x] = \text{iterate}[	ext{plus}[x], \{0\}] 
\]

Division is defined by rotating \( \text{id}[	ext{complement}\{0\}] \circ \text{NATMUL} \).

Multiplicative law of exponents: if \( y \in \omega \) then
\[
\text{power}[	ext{image}[	ext{power}[x], \{y\}]] = \text{power}[x] \circ \text{NATMUL} \circ \text{LEFT}[y].
\]
Constructing the set $\mathbb{Z}$ of integers.

\[ +0 = -0 = \text{id}[\omega] \]
\[ +1 = \text{id}[\omega] \circ \text{SUCC} \]
\[ +2 = \text{id}[\omega] \circ \text{SUCC} \circ \text{SUCC} \]
\[ +n = \text{NATADD} \circ \text{LEFT}[n] \]
\[ -n = \text{inverse}[+n] \]
Reification

Gödel’s algorithm for class formation provides a mechanical way to eliminate quantifiers over set variables. An alternative method was discovered that is sometimes faster. The idea is to associate with each class constructor \( F[x] \) a relation \( \text{reify}[x, F[x]] \) that captures information about how the constructor acts on sets.

\[
\text{reify}[x, F[x]] = \{ \langle x, y \rangle \mid y \in F[x] \}
\]

For each class constructor \( G \) definable from Gödel’s primitives there is a formula for the reification of composite constructors \( G[F[x]] \) in terms of the reification of \( F \). For example:

\[
\text{reify}[x, \text{complement}[F[x]]] = \text{Id} \circ \text{complement}[\text{reify}[x, F[x]]]
\]

\[
\text{reify}[x, \text{domain}[F[x]]] = \text{FIRST} \circ \text{reify}[x, F[x]]
\]

\[
\text{reify}[x, \{F[x]\}] = \text{VERTSECT}[\text{reify}[x, F[x]]]
\]

\[
\text{reify}[x, F[x] \cap G[x]] = \text{reify}[x, F[x]] \cap \text{reify}[x, G[x]]
\]

The reification does not capture all information in a constructor; one cannot recover a constructor from its reification.
Conclusions

1. The uniqueness theorem for iteration can serve as a substitute for mathematical induction in many applications, including the development of the arithmetic of natural numbers, the theory of transitive closures of relations, the Schröder-Bernstein theorem, and so on.

2. Proving theorems involving iterative definitions does not require modifying the theorem prover itself. Any good first order theorem prover like Otter should be adequate for proofs with iterative constructions. All one needs is an appropriate set of definitions and to derive a body of facts about iteration in the form of clauses that can be made available for applications of interest.

3. The GOEDEL program is useful for finding needed definitions, as well as helping to discover useful lemmas.