CORE and HULL Constructors in Gödel's Class Theory

Johan G. F. Belinfante and Tiffany D. Goble *

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

Abstract. The `GOEDEL` program, a computer implementation of Gödel's algorithm for class formation in Mathematica$^{\text{TM}}$ was used for formulating definitions and discovering theorems about topology and its generalizations, working within Gödel's class theory. A general characterization of CORE[x] and HULL[x] functions discovered in the course of this work is the primary focus of this paper.

1 Introduction

Computers are not only valuable for automated reasoning and for formal verification in mathematics, but can also contribute significantly to the formulation of definitions, simplifying the statements and the proofs of theorems, finding generalizations, and can sometimes even lead to the discovery of new theorems. This is true in part because the very fact that one is using a computer will suggest natural questions that otherwise may not have been considered, and also in part because computers think in ways that are distinctly non-human. Larry Wos has aptly expressed this in the following words:

> "The human mind will never be replaced, ... but the advantage of computers is their utter lack of preconceptions. They can follow paths that are totally counterintuitive." (Chang, [2004])

Since set theory can be relied upon to formulate practically everything of interest in modern mathematics, it is arguably worthwhile to expend the considerable effort needed to develop a substantial body of standard mathematical facts which can serve as a foundation for automated reasoning involving set theory, building on Robert Boyer's seminal observation that automated reasoning in set theory can be performed within first order logic by using Kurt Gödel's reformulation of the von Neumann-Bernays axioms.

The first author's `GOEDEL` program is used for formal verification and McCune's `Otter` program is used for automatic proof search. The intention is to use the `GOEDEL` program primarily to discover how to formulate definitions and theorems, and to explore what needs to be proved, and then later to go back and find clean proofs of these results using `Otter`. Since the `GOEDEL` program itself does not produce explicit proofs, results obtained with this program will be called derivations rather than proofs, even though such derivations often are in fact more detailed than what passes for a proof in common parlance. A current version of the `GOEDEL` program and a large number of sample notebooks illustrating its use are available on the first author's website:

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

---

* This research was supported on NSF ITR contract DMS 0312618.

In the past year, some preliminary successes were achieved by the authors using the `GOEDEL` computer program to derive basic theorems of point-set topology and its generalizations. This work was in part inspired by a remarkable paper by McCune and Wick ([1989]) in which `Otter` was used to prove some theorems of point-set topology before a completely adequate set-theoretic basis was available. Another early effort (Farmer [1991]) to apply automated reasoning to topology produced proofs of two theorems in metric-space topology. Considerable efforts have also been made (Bancerek, [1997]) using `Mizar` to formalize topological notions for the purpose of computer verification. There have undoubtedly been other pioneering efforts to apply computers to reasoning in topology, and it is sincerely hoped that there will be many more in the future.

Unlike `Otter`, the `GOEDEL` program is not an automated reasoning program, and does not produce explicit proofs, but it does contain numerous rewrite rules for simplifying descriptions of classes and assertions about them, and there is a (fairly primitive) construct called `SubstTest` that can be used to carry out deductions by hand. This is one of several tools that permit one to use existing rewrite rules to deduce new rewrite rules, thereby providing a means for the program to gradually evolve into an ever increasingly powerful reasoning assistant. Some theorems about the T1 and T2 separation axioms, the cofinite topology, and compactness were among the results that were derived using `GOEDEL`. Further details about these and other applications to topology can be found on the first author's website. This paper will focus on just one aspect of this ongoing research, a generalization of Kuratowski's characterization of topologies via closure operators that was discovered in the course of our work.

Because our work is being done within the framework of the Gödel-Bernays class theory, the collections of sets that can be considered do not have to be sets. The axiom of regularity and the axiom of choice are not assumed to hold unless explicitly mentioned. Generalizing topological ideas to proper classes is not just an idle pastime. In ordinal number theory, for example, an important concept is that of a full (or transitive) class. A class is *full* if all its members are subsets. The class `FULL` of all full sets is technically not a topology because it is a proper class, but it shares some properties of a topology. In particular, the class of all full sets is closed under arbitrary unions, and under arbitrary intersections, too. The class `H[FINITE]` of hereditarily finite sets, for example, can be characterized as the interior of the class `FINITE` of finite sets with respect to this pseudotopology.

The needs of automated reasoning raise many mathematical questions that are not readily answered in the standard mathematical literature. This is particularly true in situations where the standard literature deals only with the case of sets and is silent on the issue whether sethood is really necessary. Some of the issues requiring attention are just details that can be readily resolved. For example, upon proving a standard result of the form $\mathtt{A} \Rightarrow \mathtt{B}$, one might wish to add a rewrite rule that automatically rewrites `A` to `B`. Doing so would only be justified if the converse implication also holds, forcing one therefore to think about whether the converse is true even in situations where the converse statement might not otherwise be particularly interesting. Sometimes, however, genuinely interesting issues are brought to one's attention in this fashion. In the course of the research reported in this paper, a number of interesting mathematical questions arose for which the authors were unable to provide entirely satisfactory resolutions. Some of these open questions will be mentioned as we go along.

2 Notation

In the Gödel class theory, equality and the membership predicate $\in$ are taken as primitive undefined concepts, subject to various axioms, and everything is a class. By definition, a class is a *set* if there is a class to which it belongs. Classes that are not sets are called *proper classes*. The argument leading to Russell's paradox can be used to show that Russell's class of all sets that do not belong to themselves is a proper class. The axiom of replacement implies that any subclass of a set is a set, and from this one can deduce that the class `V` of all sets must also be a proper class. Another familiar example of a proper class is the class Ω of all ordinals.

Sets are sometimes called *small* classes, the idea being that all the known examples of proper classes are extremely large. This notion is reinforced by the axiom of replacement which implies that any subclass of a set is a set. But it is somewhat embarrassing that one can not even show for example that every proper class contains an infinite subset. This open question came up in the course of dealing with theorems about finite topological spaces. The natural question arose whether one could simply add a rewrite rule that transforms the statement $\mathtt{P}[\mathtt{x}] \subset \mathtt{FINITE}$ to $\mathtt{x} \in \mathtt{FINITE}$. These statements are of course trivially equivalent when `x` is a set because any set belongs to its power set, but to avoid a conditional rewrite rule, one really wants to know whether sethood is needed. The general case amounts to the question whether a proper class must always contain an infinite subset. (A similar question, whether an infinite set must contain a countable subset, is known to require an application of the axiom of choice.)

Because the Zermelo-Fraenkel version of set theory is the one most familiar to most mathematicians, proper classes are used rather sparingly in the literature. Even authors (Rubin [1967], Mendelson [1987]) that do embrace the NBG axioms for class theory generally mention proper classes only to convince the reader that the paradoxes of naive set theory are resolved, but rarely take full advantage of constructions involving proper classes. Since Gödel's algorithm (Gödel, [1940]) routinely produces such constructions, proper classes feature prominently in our work. For this reason it is appropriate to review some basic proper classes that will be used in the sequel.

In addition to the proper class `V`, there is another proper class, the membership relation, whose existence is postulated by one of the axioms in Gödel's class theory. This axiom asserts that there is a class `E` whose members are all ordered pairs $\langle \mathtt{x}, \mathtt{y} \rangle$ satisfying $\mathtt{x} \in \mathtt{y}$.

The class `image[x,y]` is defined as the range of the restriction of the relation `x` to the class `y`. The *vertical section* of a relation `x` at a set `y` is $\mathtt{image}[\mathtt{x}, \{\mathtt{y}\}]$, where $\{\mathtt{y}\}$ denotes the singleton of `y`. A relation `x` is *thin* if `image[x,y]` is a set whenever `y` is set. Any set, of course, is thin, and the axiom of replacement is equivalent to the assertion that all functions are thin. It can be shown that `x` is thin if all its vertical sections are sets.

The class `ub[x,y]` of upper bounds of a class `y` with respect to a relation `x` is defined to be the class

$$\mathtt{ub}[\mathtt{x}, \mathtt{y}] = \mathtt{image}[\mathtt{x}', \mathtt{y}]\,',$$

where $\mathtt{x}'$ denotes the complement of the class `x`. Similarly, the class `lb[x,y]` of lower bounds is defined by replacing `x` with its inverse:

$$\mathtt{lb}[\mathtt{x}, \mathtt{y}] = \mathtt{ub}[\mathtt{inverse}[\mathtt{x}], \mathtt{y}].$$

An important application of this is the formula for the intersection of a collection of sets,

$$\mathtt{lb}[\mathtt{E},\mathtt{x}] = \bigcap \mathtt{x}.$$

The sum class $\bigcup \mathtt{x}$ and the power class `P[x]` of a class `x` can be defined in terms of the membership relation by the formulas

$$\bigcup \mathtt{x} = \mathtt{image}[\mathtt{inverse}[\mathtt{E}],\mathtt{x}]$$

and

$$\mathtt{P}[\mathtt{x}] = \mathtt{image}[\mathtt{E}',\mathtt{x}]\,'.$$

According to the sum class and power set axioms, these are sets when `x` is a set. It follows from the sum class axiom that the inverse of the membership relation is thin.

For any class `x`, the upper bound relation

$$\mathtt{UB}[\mathtt{x}] = (\mathtt{x}' \circ \mathtt{inverse}[\mathtt{E}])\,' \cap (\mathtt{V} \times \mathtt{V})$$

is the class of ordered pairs $\langle \mathtt{y},\mathtt{z}\rangle$ such that `z = ub[x,y]`. The lower bound relation is defined by `LB[x] = UB[inverse[x]]`. In particular, the subset relation `S = UB[E]` is the class of ordered pairs $\langle \mathtt{y},\mathtt{z}\rangle$ such that $\mathtt{y} \subset \mathtt{z}$. It follows from the power set axiom that the inverse of the subset relation is thin.

The identity relation is $\mathtt{Id} = \mathtt{S} \cap \mathtt{inverse}[\mathtt{S}]$. The restriction of the identity relation to a class `x` is denoted by $\mathtt{id}[\mathtt{x}]$. The class `fix[x]` of all fixed points of `x`, defined by

$$\mathtt{fix}[\mathtt{x}] = \{\mathtt{y} \mid \langle \mathtt{y},\mathtt{y}\rangle \in \mathtt{x}\},$$

is related to the identity relation by

$$\mathtt{x} \cap \mathtt{Id} = \mathtt{id}[\mathtt{fix}[\mathtt{x}]].$$

3 Eliminating set variables

All statements in mathematics can be automatically converted by means of Gödel's algorithm into equations without set-variables, something which Alfred Tarski and Steven Givant ([1987]) had shown could be done in theory, on the basis of a calculus of relations. This result is not limited however to the special formalism that they consider, but can also be achieved in the more traditional setting of NBG class theory. When this process (called `assert` in the `GOEDEL` program) is applied to a statement containing quantifiers over set-variables, the statement is converted into a logically equivalent equation without quantifiers. In the current version of the `GOEDEL` program, the equations that one gets are often subsequently converted by rewrite rules to simpler statements of a non-equational nature. For example, the axiom of regularity is transformed into the statement that the universal class is the only class that contains its own power class. It should be noted that this technique cannot be used to eliminate quantifiers over class variables.

A recent major discovery provides another method for eliminating set variables, using a process that has been named reification (Belinfante, [2003]). The idea is to associate to each constructor `f[x]` in the Gödel class theory the relation `R` of all ordered pairs $\langle \mathtt{x}, \mathtt{y} \rangle$ such that `y` belongs to the class `f[x]`,

$$\mathtt{R} = \mathtt{reify}[\mathtt{x}, \mathtt{f}[\mathtt{x}]] = \{\langle \mathtt{x}, \mathtt{y} \rangle \mid \mathtt{y} \in \mathtt{f}[\mathtt{x}]\}.$$

For each constructor `f`, there is a formula expressing the reification of composite constructors `f[g[x]]` in terms of the reification of the inner constructor `g`. These reification rules can often serve as a substitute for Gödel's algorithm for eliminating set variables, with improved execution time and cleaner output. Some examples that illustrate this technique will be presented in this paper.

In many cases the result of eliminating variables produces formulas involving function constructions. To each relation `x` there corresponds a function `VERTSECT[x]` which assigns to each set `y` the vertical section of `x` at `y` whenever the vertical section is a set. This important constructor can be used to define many important special functions. For example, the function `POWER = VERTSECT[inverse[S]]` assigns to each set its power set, the function `SINGLETON = VERTSECT[Id]` assigns to each set its singleton, and the function

$$\mathtt{BIGCAP} = \mathtt{VERTSECT}[\mathtt{LB}[\mathtt{E}]]$$

takes any nonempty collection of sets `x` to its intersection $\bigcap \mathtt{x}$. A relation `x` is thin if `domain[VERTSECT[x]] = V`. The relation `LB[E]` fails to be thin because its vertical section at the empty set is the proper class `V`; the domain of `BIGCAP` is the class $\{\mathtt{0}\}'$ of nonempty sets. In principle, any function `f` can be expressed as a `VERTSECT`,

$$\mathtt{f} = \mathtt{VERTSECT}[(\mathtt{inverse}[\mathtt{E}] \circ \mathtt{f}) \cup (\mathtt{domain}[\mathtt{f}]' \times \mathtt{V})].$$

This permits one to construct `f` whenever formulas for $\mathtt{inverse}[\mathtt{E}] \circ \mathtt{f}$ and `domain[f]` are available.

A closely related constructor for functions is

$$\mathtt{IMAGE}[\mathtt{x}] = \mathtt{VERTSECT}[\mathtt{x} \circ \mathtt{inverse}[\mathtt{E}]],$$

which assigns to each set `y` the image `image[x,y]` provided the latter is a set. For example, the function `BIGCUP = IMAGE[inverse[E]]` assigns to each set its sum class, the function `IMAGE[id[x]]` assigns to each set its intersection with `x`, and the function `IMAGE[SWAP]` takes `x` to `inverse[x]`. For this last function, it is generally more convenient to work with its restriction to the class $\mathtt{P}[\mathtt{V} \times \mathtt{V}]$ of all small relations,

$$\mathtt{INVERSE} = \mathtt{IMAGE}[\mathtt{SWAP}] \circ \mathtt{id}[\mathtt{P}[\mathtt{V} \times \mathtt{V}]],$$

because this restriction is one-to-one; indeed, this restriction is an involution, that is, a function which is equal to its own inverse,

$$\mathtt{inverse}[\mathtt{INVERSE}] = \mathtt{INVERSE}.$$

In a later section, an application of this function to integer arithmetic will be described.

The domain of `IMAGE[x]` is

$$\mathtt{domain}[\mathtt{IMAGE}[\mathtt{x}]] = \mathtt{P}[\mathtt{domain}[\mathtt{VERTSECT}[\mathtt{x}]]].$$

In particular, when `x` is thin, the domains of both `VERTSECT[x]` and `IMAGE[x]` are equal to the universal class `V`. One of the most-used rewrite rules in the `GOEDEL` program transforms $\mathtt{x} \circ \mathtt{inverse}[\mathtt{E}]$ to $\mathtt{inverse}[\mathtt{E}] \circ \mathtt{IMAGE}[\mathtt{x}]$ whenever `x` is thin. This rewrite rule in effect automatically replaces thin relations with functions.

While any function can be written as `VERTSECT[y]`, not all functions can be written in the form `IMAGE[y]`. For the special case that `y` is thin, the `GOEDEL` program yields a simple characterization of such functions. In one direction, one has

$$(\mathtt{x} = \mathtt{IMAGE}[\mathtt{y}] \ \ \& \ \ \mathtt{thin}[\mathtt{y}]) \ \Rightarrow \ \mathtt{BIGCUP} \circ \mathtt{IMAGE}[\mathtt{x}] = \mathtt{x} \circ \mathtt{BIGCUP}.$$

Conversely, if $\mathtt{BIGCUP} \circ \mathtt{IMAGE}[\mathtt{x}] = \mathtt{x} \circ \mathtt{BIGCUP}$, and $\mathtt{x} \subset \mathtt{V} \times \mathtt{V}$, then $\mathtt{x} = \mathtt{IMAGE}[\mathtt{y}]$, where $\mathtt{y} = \mathtt{inverse}[\mathtt{E}] \circ \mathtt{x} \circ \mathtt{SINGLETON}$ and `y` is thin.

4 Definitions of core and hull

The class `core[x,y]` is defined to be the union of all sets that belong to `x` and are contained in `y`.

$$\mathtt{core}[\mathtt{x}, \mathtt{y}] = \bigcup (\mathtt{x} \cap \mathtt{P}[\mathtt{y}]).$$

If `x` is the collection of open sets for a topological space, and if `y` is a subset of the space, then `core[x,y]` is the interior of `y`. This concept is also of interest when `x` and `y` are proper classes. For example the class of hereditarily finite sets is `core[FULL,FINITE]`. In general, the class `H[x] = core[FULL,x]` is the largest full subclass of `x`.

The class `hull[x,y]` is defined to be the intersection of all sets that belong to `x` and which contain `y`.

$$\mathtt{hull}[\mathtt{x}, \mathtt{y}] = \bigcap (\mathtt{x} \cap \mathtt{image}[\mathtt{S}, \{\mathtt{y}\}]).$$

For example, if `x` is the collection of closed sets of a topological space, and if `y` is a subset of the space, then `hull[x,y]` is the closure of `y`.

The formal similarity between the definitions of `core` and `hull` breaks down when proper classes are considered. Since there are no sets that contain a proper class, one has `hull[x,y] = V` whenever `y` is a proper class. This circumstance has practical repercussions. For example, for every class `x` there is a smallest class `tc[x]` which is full and contains `x`. When `x` is a set, the transitive closure is given by the simple formula `tc[x] = hull[FULL,x]`, but when `x` is a proper class, a slightly different construction is required, based on the intuitive notion that a proper class can be approximated in some sense by very large subsets. To make this vague idea more precise, some additional definitions will be needed, which will now be explained.

A function `f` is *idempotent* if $\mathtt{f} \circ \mathtt{f} = \mathtt{f}$. If a function is idempotent, then every element of its range is a fixed point: `range[f] = fix[f]`. It is convenient to introduce two families of idempotent functions `CORE[x]` and `HULL[x]` each depending on a parameter `x` which in principle can be any class. The function `CORE[x]` can be formally characterized as the class of all ordered pairs $\langle \mathtt{y}, \mathtt{z} \rangle$ such that `z = core[x,y]` and the function `HULL[x]` is the class of pairs $\langle \mathtt{y}, \mathtt{z} \rangle$ such that `z = hull[x,y]`. If `x` is the collection of open sets for a topological space, then the restriction of `CORE[x]` to the class of all subsets of the topological space is the interior operator. If `x` is the collection of closed sets for a topological space, then the restriction of `HULL[x]` to the class of all subsets of the topological space is the closure operator. Kuratowski characterized these closure operators and showed that a topology is uniquely determined by its closure operator.

Despite the similarity of the characterizations of the functions `CORE[x]` and `HULL[x]`, some of their properties are actually quite different, and this already shows up in the formulas used to define them. Since class formation is part of the metatheory of Gödel's class theory, and not part of the theory itself, the similarity of the characterizations of these functions in terms of requirements for ordered pairs to belong to them is somewhat misleading. Gödel's algorithm yields an equational definition for each of these functions. In the case of `CORE[x]` one obtains

$$\mathtt{CORE}[\mathtt{x}] = \mathtt{BIGCUP} \circ \mathtt{IMAGE}[\mathtt{id}[\mathtt{x}]] \circ \mathtt{POWER}.$$

For `HULL[x]` one finds a rather different formula, namely,

$$\mathtt{HULL}[\mathtt{x}] = \mathtt{VERTSECT}[(\mathtt{inverse}[\mathtt{E}]\,' \circ \mathtt{id}[\mathtt{x}] \circ \mathtt{S})\,'].$$

It is probably worth pointing out that this formula does not actually appear explicitly in the `GOEDEL` program, the reason being that from this definition one can derive the formula

$$\mathtt{inverse}[\mathtt{E}]\,' \circ \mathtt{id}[\mathtt{x}] \circ \mathtt{S} = \mathtt{inverse}[\mathtt{E}]\,' \circ \mathtt{HULL}[\mathtt{x}].$$

The latter formula occurs as a rewrite rule that subsumes the preceding formula. A similar result holds, by the way, for `CORE[x]`,

$$\mathtt{inverse}[\mathtt{E}] \circ \mathtt{id}[\mathtt{x}] \circ \mathtt{inverse}[\mathtt{S}] = \mathtt{inverse}[\mathtt{E}] \circ \mathtt{CORE}[\mathtt{x}].$$

Further dissimilarities are found in other properties of these functions. Consider for example the domains of these functions. When `y` is a set, so is its power set `P[y]`. Since any subclass of a set is a set, the intersection $\mathtt{x} \cap \mathtt{P}[\mathtt{y}]$ is a set, and hence, by the sum class axiom, $\mathtt{z} = \mathtt{core}[\mathtt{x}, \mathtt{y}] = \bigcup(\mathtt{x} \cap \mathtt{P}[\mathtt{y}])$ is a set. Consequently, the domain of `CORE[x]` is the class `V` of all sets. On the other hand, the intersection of a collection of sets is a set if and only if that collection is not empty. Consequently, the domain of `HULL[x]` is the class of all subsets of members of `x`,

$$\mathtt{domain}[\mathtt{HULL}[\mathtt{x}]] = \mathtt{image}[\mathtt{inverse}[\mathtt{S}], \mathtt{x}].$$

By the axiom of replacement, it follows from this that while the function `CORE[x]` is always a proper class, the function `HULL[x]` is a set if and only if `x` is a set.

The function `TC = HULL[FULL]` provides a method to construct the transitive closure of any class,

$$\mathtt{tc}[\mathtt{x}] = \bigcup \mathtt{image}[\mathtt{TC}, \mathtt{P}[\mathtt{x}]].$$

In practice one also needs an additional formula,

$$\mathtt{tc}[\mathtt{x}] = \mathtt{range}[\mathtt{iterate}[\mathtt{inverse}[\mathtt{E}], \mathtt{x}]],$$

which allows one to use induction to derive the properties of the transitive closure. Both of these formulas hold for any class `x`, not just for sets.

Another important example is the family of functions

$$\mathtt{ADJOIN}[\mathtt{x}] = \mathtt{HULL}[\mathtt{image}[\mathtt{S}, \{\mathtt{x}\}]].$$

If `x` is a set, this is a total function which takes any set `y` to the set $\mathtt{x} \cup \mathtt{y}$. When `x` is not a set, the function is the empty set.

In general, the relation `iterate[x, y]` can be characterized (Belinfante [2003]) by the conditions that its vertical section at the empty set is the class `y`, and for each natural number `n`, the image under `x` of the vertical section at `n` produces the vertical section at the successor of `n`. Explicitly, the following uniqueness theorem holds for iteration:

$$\mathtt{z} \circ \mathtt{SUCC} = \mathtt{x} \circ \mathtt{z} \;\;\&\;\; \mathtt{image}[\mathtt{z}, \{\mathtt{0}\}] = \mathtt{y} \;\;\Rightarrow\;\; \mathtt{iterate}[\mathtt{x}, \mathtt{y}] = \mathtt{z} \circ \mathtt{id}[\omega],$$

where $\omega = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots\}$ denotes the set of all natural numbers, and `SUCC` denotes the successor function, which takes any set `x` to its successor $\mathtt{succ}[\mathtt{x}] = \mathtt{x} \cup \{\mathtt{x}\}$.

5 Open questions concerning Uclosure and Aclosure

The `Uclosure` of any class `x` is the class of all unions of subsets of `x`,

$$\mathtt{Uclosure}[\mathtt{x}] = \mathtt{image}[\mathtt{BIGCUP}, \mathtt{P}[\mathtt{x}]].$$

A familiar application of this is the construction of a topology from a topological base. Similarly, `Aclosure[x]` is the class of all intersections of subsets of `x`,

$$\mathtt{Aclosure}[\mathtt{x}] = \mathtt{image}[\mathtt{BIGCAP}, \mathtt{P}[\mathtt{x}]].$$

Every class `x` is contained in its own Aclosure and Uclosure. A class `x` is closed under arbitrary intersections if it satisfies `Aclosure[x] = x`, and is closed under arbitrary unions if `Uclosure[x] = x`. This is the case, for example, for the class `invar[x]` of all sets invariant under an operation `x`,

$$\mathtt{invar}[\mathtt{x}] = \{\mathtt{y} \mid \mathtt{image}[\mathtt{x}, \mathtt{y}] \subset \mathtt{y}\}.$$

Important examples include the class `FULL = invar[inverse[E]]`, and any power class $\mathtt{P}[\mathtt{x}] = \mathtt{invar}[\mathtt{x}' \times \mathtt{V}]$. Further examples can be constructed by using formulas such as

$$\mathtt{invar}[\mathtt{x} \cup \mathtt{y}] = \mathtt{invar}[\mathtt{x}] \cap \mathtt{invar}[\mathtt{y}].$$

The idempotent functions `UCLOSURE` and `ACLOSURE` take a set `x` to `Uclosure[x]` and `ACLOSURE`, respectively:

$$\mathtt{ACLOSURE} = \mathtt{IMAGE}[\mathtt{BIGCAP}] \circ \mathtt{POWER},$$
$$\mathtt{UCLOSURE} = \mathtt{IMAGE}[\mathtt{BIGCUP}] \circ \mathtt{POWER}.$$

Each of these functions is a `HULL` function:

$$\mathtt{ACLOSURE} = \mathtt{HULL}[\mathtt{fix}[\mathtt{ACLOSURE}]],$$
$$\mathtt{UCLOSURE} = \mathtt{HULL}[\mathtt{fix}[\mathtt{UCLOSURE}]].$$

It is currently not known whether the functions `UCLOSURE` and `ACLOSURE` commute. At issue here is whether the distributive law extends to infinite unions and intersections; can an infinite union of infinite intersections be rewritten as an infinite intersection of infinite unions, and vice versa? A closely related question is whether `fix[ACLOSURE]` is invariant under `UCLOSURE`, and conversely.

Since the functions `CORE[x]` and `HULL[x]` are both idempotent, their ranges and fixed point classes are equal. In the case of `CORE[x]`, one has

$$\texttt{Uclosure}[\texttt{x}] = \texttt{range}[\texttt{CORE}[\texttt{x}]] = \texttt{fix}[\texttt{CORE}[\texttt{x}]],$$

but for the case of `HULL[x]` all that has been proved at this point is that

$$\texttt{Aclosure}[\texttt{x}] \subset \texttt{range}[\texttt{HULL}[\texttt{x}]] = \texttt{fix}[\texttt{HULL}[\texttt{x}]].$$

It remains an open question whether the class `fix[HULL[x]]` is in fact always equal to the class `Aclosure[x]`. Equality has been proved for the important special case that `x` is a set, and also for various special proper classes. It would be desirable to have either a general proof that these classes are always equal or else a counterexample if they are not always equal. Another open quesion is whether the constructor `Aclosure` is idempotent in general, as is known to be the case for `Uclosure`. If `x` is a set, one has

$$\texttt{Aclosure}[\texttt{Aclosure}[\texttt{x}]] = \texttt{Aclosure}[\texttt{x}],$$

which suffices to show that the function `ACLOSURE` is idempotent. Incidentally, the closely related operation `fix[HULL[x]]` is known to be idempotent for arbitrary classes; in fact,

$$\texttt{HULL}[\texttt{fix}[\texttt{HULL}[\texttt{x}]]] = \texttt{HULL}[\texttt{x}].$$

The functions `CORE[x]` and `HULL[x]` satisfy the equations

$$\begin{aligned}\texttt{CORE}[\texttt{Uclosure}[\texttt{x}]] &= \texttt{CORE}[\texttt{x}],\\ \texttt{HULL}[\texttt{Aclosure}[\texttt{x}]] &= \texttt{HULL}[\texttt{x}].\end{aligned}$$

Thus, for example, it makes no difference if one replaces a topology by a base for the topology in defining interiors.

One of the more fascinating properties of the function `UCLOSURE` is that it commutes with `IMAGE[IMAGE[x]]` for any class `x`. A special case of this was used in the study of relative topologies.

6 Characterizing CORE and HULL

It is convenient to express various strong versions of monotonicity in terms of the subset relation `S`. A function `f` is *monotone* if

$$\texttt{f} \circ \texttt{S} \circ \texttt{inverse}[\texttt{f}] \subset \texttt{S}.$$

This quantifier-free statement is equivalent to the condition

$$\forall \texttt{u}, \texttt{v}, \texttt{x}, \texttt{y}\ ((\langle \texttt{u}, \texttt{x}\rangle \in \texttt{f}\ \ \&\ \ \langle \texttt{v}, \texttt{y}\rangle \in \texttt{f}\ \ \&\ \ \texttt{u} \subset \texttt{v}) \Rightarrow \texttt{x} \subset \texttt{y}).$$

Similarly, a function is *antitone* if

$$\mathtt{f} \circ \mathtt{inverse}[\mathtt{S}] \circ \mathtt{inverse}[\mathtt{f}] \subset \mathtt{S}.$$

A function is *total* if `domain[f] = V`. A class `x` is *hereditary* if every subset of a member is a member, that is, if `image[inverse[S], x] = x`. It will be said that `x` and `y` *subcommute* if $\mathtt{x} \circ \mathtt{y} \subset \mathtt{y} \circ \mathtt{x}$. Any relation that subcommutes with `S` has a hereditary domain. Any monotone function with a hereditary domain subcommutes with the subset relation, and conversely, any function that subcommutes with `S` is monotone. In other words, the condition that a function `f` subcommutes with `S` is equivalent to saying that it is not only monotone, but its domain is hereditary. A total monotone function `f` satisfies the even stronger condition that $\mathtt{S} \subset \mathtt{inverse}[\mathtt{f}] \circ \mathtt{S} \circ \mathtt{f}$, and conversely, this condition implies that `f` is monotone and total. A total function which subcommutes with the subset relation also subcommutes with the inverse of the subset relation.

For example, the domain of the monotone function `IMAGE[x]` is a power class,

$$\mathtt{domain}[\mathtt{IMAGE}[\mathtt{x}]] = \mathtt{P}[\mathtt{domain}[\mathtt{VERTSECT}[\mathtt{x}]]],$$

and is therefore hereditary. Consequently `IMAGE[x]` subcommutes with `S`. The function `IMAGE[x]` is total when `x` is thin; `IMAGE[x]` subcommutes with `inverse[S]` if and only if `x` is thin. This fact had been proved in one direction using `Otter`. The discovery that the converse holds was motivated by the desire to avoid a conditional rewrite rule in the `GOEDEL` program; conditional rewrite rules slow the program down significantly. Some additional facts did require adding conditional rewrite rules: If `x` is a function, then `IMAGE[x]` commutes with `S`, and if `x` is a total one-to-one function, then `IMAGE[x]` commutes with `S`.

The idempotent functions `CORE[x]` and `HULL[x]` are both monotone, and both of them subcommute with the subset relation because their domains are hereditary. The function `CORE[x]`, of course, has the stronger property of being total, and therefore subcommutes also with `inverse[S]`, whereas this is generally not the case for `HULL[x]`. The other important difference between these functions is that `HULL[x]` is contained in `S`, whereas `CORE[x]` is contained in `inverse[S]`.

These properties characterize the functions `CORE[x]` and `HULL[x]`. If a monotone idempotent function has a hereditary domain and is contained in the subset relation, then it is a `HULL` function,

$$(\mathtt{f} \circ \mathtt{f} = \mathtt{f} \ \ \& \ \ \mathtt{FUNCTION}[\mathtt{f}] \ \ \& \ \ \mathtt{f} \subset \mathtt{S} \ \ \& \ \ \mathtt{f} \circ \mathtt{S} \subset \mathtt{S} \circ \mathtt{f}) \Rightarrow \mathtt{f} = \mathtt{HULL}[\mathtt{fix}[\mathtt{f}]].$$

If a total monotone idempotent function is contained in the inverse of the subset relation, then it is a `CORE` function,

$$\begin{aligned}(&\mathtt{f} \circ \mathtt{f} = \mathtt{f} \ \ \& \ \ \mathtt{FUNCTION}[\mathtt{f}] \ \ \& \ \ \mathtt{domain}[\mathtt{f}] = \mathtt{V} \\ &\ \ \& \ \ \mathtt{f} \circ \mathtt{S} \subset \mathtt{S} \circ \mathtt{f} \ \ \& \ \ \mathtt{f} \subset \mathtt{inverse}[\mathtt{S}]) \quad \Rightarrow \quad \mathtt{f} = \mathtt{CORE}[\mathtt{fix}[\mathtt{f}]].\end{aligned}$$

The interiors and closures of subsets of a topological space are related via relative complementation. The result does not depend on the topology axioms. A general result can be derived using the above characterization of `HULL` functions. For any set `x` there is a relative complementation function `RC[x]` consisting of all ordered pairs $\langle \mathtt{y}, \mathtt{z} \rangle$ such that $\mathtt{y} \cup \mathtt{z} = \mathtt{x}$ and $\mathtt{y} \cap \mathtt{z} = \mathtt{0}$.

$$\mathtt{RC}[\mathtt{x}] = \mathtt{DISJOINT} \cap \mathtt{image}[\mathtt{inverse}[\mathtt{CUP}], \{\mathtt{x}\}].$$

This function is antitone, and is its own inverse. Note that `RC[x] = 0` when `x` is a proper class. Since the function `CORE[y]` is monotone, the composite function $\mathtt{RC}[\mathtt{x}] \circ \mathtt{CORE}[\mathtt{z}] \circ \mathtt{RC}[\mathtt{x}]$ is monotone. The other conditions in the characterization of a `HULL` function also hold, and so, applying the characterization of `HULL` to this special case yields the formula

$$\mathtt{RC}[\mathtt{x}] \circ \mathtt{CORE}[\mathtt{y}] \circ \mathtt{RC}[\mathtt{x}] = \mathtt{HULL}[\mathtt{image}[\mathtt{RC}[\mathtt{x}], \mathtt{Uclosure}[\mathtt{y}]]].$$

Actually, a slightly more general result holds:

$$\mathtt{HULL}[\mathtt{image}[\mathtt{RC}[\mathtt{x}], \mathtt{y}]] = \mathtt{RC}[\mathtt{x}] \circ \mathtt{CORE}[\mathtt{y}] \circ \mathtt{id}[\mathtt{image}[\mathtt{S}, \mathtt{y}]] \circ \mathtt{RC}[\mathtt{x}].$$

The extra factor `id[image[S,y]]` is not needed when $\mathtt{0} \in \mathtt{y}$. For the record, we note that this more general formula had in fact been derived by an application of reification before the characterization of `HULL` had been established.

7 Applications to Topology

A *topology* is a set `t` of sets which is closed under binary intersections and arbitrary unions, that is, a set satisfying

$$\mathtt{t} = \mathtt{Uclosure}[\mathtt{t}],$$
$$\mathtt{t} = \mathtt{image}[\mathtt{CAP}, \mathtt{t} \times \mathtt{t}].$$

Any power set is a topology, as are all successor ordinals. In general, the Uclosure of an ordinal number is the successor of its sum class. The *cofinite topology* for any set `x` holds the empty set, and all subsets of `x` whose relative complement in `x` is finite,

$$\mathtt{Uclosure}[\mathtt{image}[\mathtt{RC}[\mathtt{x}], \mathtt{FINITE}]] = \{\mathtt{0}\} \cup \mathtt{image}[\mathtt{RC}[\mathtt{x}], \mathtt{FINITE}].$$

In general, the class `binclosed[x]` of sets closed under a binary operation `x` is

$$\mathtt{binclosed}[\mathtt{x}] = \{\mathtt{t} \mid \mathtt{image}[\mathtt{x}, \mathtt{t} \times \mathtt{t}] \subset \mathtt{t}\} = \mathtt{fix}[\mathtt{S} \circ \mathtt{IMAGE}[\mathtt{x}] \circ \mathtt{CART} \circ \mathtt{DUP}].$$

Using this notation, one can write the class `TOPS` of all topologies as the intersection

$$\mathtt{TOPS} = \mathtt{fix}[\mathtt{UCLOSURE}] \cap \mathtt{binclosed}[\mathtt{CAP}].$$

Note the resemblance of `binclosed[x]` with `invar[x]`. In fact, `invar[x]` can even be written as $\mathtt{binclosed}[\mathtt{x} \circ \mathtt{inverse}[\mathtt{DUP}]]$. It should therefore not come as much of a surprise that `binclosed` shares some of the properties of `invar`. In particular, it is closed under arbitrary intersections: `Aclosure[binclosed[x]] = binclosed[x]`.

The class `TOPS` is closed under arbitrary intersections: `Aclosure[TOPS]=TOPS`. Any set `x` generates a smallest topology `hull[TOPS, x]` that contains `x`. Since the class `binclosed[CAP]` is invariant under `UCLOSURE`, it follows from the characterization of `HULL` functions that

$$\mathtt{HULL}[\mathtt{TOPS}] = \mathtt{UCLOSURE} \circ \mathtt{HULL}[\mathtt{binclosed}[\mathtt{CAP}]].$$

In other words, the topology generated by a set can be obtained in two steps; first one generates a topological base, and then one applies Uclosure to obtain the topology itself,

$$\mathtt{hull}[\mathtt{TOPS}, \mathtt{x}] = \mathtt{Uclosure}[\mathtt{hull}[\mathtt{binclosed}[\mathtt{CAP}], \mathtt{x}]].$$

Another corollary is a succinct formula `TOPS = image[UCLOSURE, binclosed[CAP]]` for the class of all topologies.

If `t` is a topology, its members are called the open subsets of the topological space $\bigcup \mathtt{t}$, and their relative complements are called the closed sets. So $\mathtt{c} = \mathtt{image}[\mathtt{RC}[\bigcup \mathtt{t}], \mathtt{t}]$ is the set of all closed sets. The interior of a subset $\mathtt{x} \subset \bigcup \mathtt{t}$ is `core[t, x]`, and its closure is `hull[c, x]`.

If `t` is any topology, and `x` is any class, then the set

$$\mathtt{image}[\mathtt{IMAGE}[\mathtt{id}[\mathtt{x}]], \mathtt{t}] = \{\mathtt{z} \mid (\exists \mathtt{y} \in \mathtt{t})\ \mathtt{z} = \mathtt{x} \cap \mathtt{y}\}$$

is also a topology. When $\mathtt{x} \subset \bigcup \mathtt{t}$ this is known as the *relative topology* on the subset `x`. For each open set $\mathtt{y} \in \mathtt{t}$, the intersection $\mathtt{z} = \mathtt{x} \cap \mathtt{y}$ is open in the relative topology. Since the variable `t` refers to a set, one can eliminate this variable, and recast the fact that the class of topologies is invariant under the process of forming relative topologies succinctly as follows:

$$\mathtt{image}[\mathtt{IMAGE}[\mathtt{IMAGE}[\mathtt{id}[\mathtt{x}]]], \mathtt{TOPS}] \subset \mathtt{TOPS}.$$

Carrying this process of variable-elimination to extremes, one could specialize the above statement to the case that `x` is any set, and then use reification to eliminate even this one remaining variable, yielding a completely variable-free statement,

$$\mathtt{image}[\mathtt{IMAGE}[\mathtt{CAP}], \mathtt{image}[\mathtt{CART}, \mathtt{range}[\mathtt{SINGLETON}] \times \mathtt{TOPS}]] \subset \mathtt{TOPS}.$$

This amounts to the assertion that `TOPS` is invariant under the relation obtained by forming the union of the functions `IMAGE[IMAGE[id[x]]]`. This is not an ordinary union because these functions are all proper classes, but one can nonetheless use the reification rules to compute such nonstandard unions. For any class constructor `f[x]`, one can compute the union of all the classes `f[x]` for which `x` is a set as follows:

$$\{\mathtt{w} \mid \exists \mathtt{x}\ \mathtt{w} \in \mathtt{f}[\mathtt{x}]\} = \mathtt{range}[\mathtt{reify}[\mathtt{x}, \mathtt{f}[\mathtt{x}]]].$$

In the present case, one needs to use the reification rules for the `IMAGE` and `id` constructors:

$$\begin{aligned}&\mathtt{reify}[\mathtt{x}, \mathtt{IMAGE}[\mathtt{y}]] = \mathtt{SWAP} \circ \mathtt{inverse}[\mathtt{rotate}[\\ &\mathtt{IMAGE}[\mathtt{rotate}[\mathtt{inverse}[\mathtt{reify}[\mathtt{x}, \mathtt{y}]]]] \circ \mathtt{CART} \circ \mathtt{SWAP}]] \circ \mathtt{SINGLETON},\\[1em] &\mathtt{reify}[\mathtt{x}, \mathtt{id}[\mathtt{y}]] = \mathtt{DUP} \circ \mathtt{reify}[\mathtt{x}, \mathtt{y}].\end{aligned}$$

In this way one readily discovers that the class `TOPS` is invariant under the relation

$$\begin{aligned}&\mathtt{range}[\mathtt{reify}[\mathtt{x},\mathtt{IMAGE}[\mathtt{IMAGE}[\mathtt{id}[\mathtt{x}]]]]] =\\ &\mathtt{IMAGE}[\mathtt{CAP}]\circ\mathtt{CART}\circ\mathtt{id}[\mathtt{range}[\mathtt{SINGLETON}]\times\mathtt{V}]\circ\mathtt{inverse}[\mathtt{SECOND}].\end{aligned}$$

Before leaving this topic, it is probably worth pointing out that even more is true; the function `id[x]` can be replaced by any one-to-one function. The class `fix[UCLOSURE]` is invariant under `IMAGE[IMAGE[x]]` for any class `x`, and the class `binclosed[CAP]` is invariant under `IMAGE[IMAGE[x]]` when `x` is any one-to-one function. From this it readily follows that

$$\mathtt{ONEONE}[\mathtt{x}]\Rightarrow\mathtt{image}[\mathtt{IMAGE}[\mathtt{IMAGE}[\mathtt{x}]],\mathtt{TOPS}]\subset\mathtt{TOPS}.$$

8 Transitive closures of relations

An important application of `HULL` functions is the theory of transitive closures of relations. This relational transitive closure `trv[x]` should not be confused with the class `tc[x]` discussed earlier.

A relation $\mathtt{x}\subset\mathtt{V}\times\mathtt{V}$ is *transitive* if $\mathtt{x}\circ\mathtt{x}\subset\mathtt{x}$. The class of all small transitive relations is denoted by `TRV`. The (relational) *transitive closure* `trv[x]` of a relation `x` is the smallest transitive relation that contains `x`. When `x` is a small relation, the transitive closure is `hull[TRV, x]`, but this construction breaks down for proper classes. If `x` is a proper class, then

$$\mathtt{trv}[\mathtt{x}]=\bigcup\mathtt{image}[\mathtt{HULL}[\mathtt{TRV}],\mathtt{P}[\mathtt{x}]]=\mathtt{image}[\mathtt{power}[\mathtt{x}],\{\mathtt{0}\}']$$

is the transitive closure of $\mathtt{x}\cap(\mathtt{V}\times\mathtt{V})$. Here `power[x]` is a relation whose vertical sections at the natural numbers are the various powers of `x`,

$$\mathtt{power}[\mathtt{x}]=\mathtt{iterate}[\mathtt{Id}\otimes\mathtt{x},\mathtt{Id}].$$

Here `cross` denotes the parallel or cross product of two relations (Belinfante, [1999]).

The situation here is actually quite similar to the theory of transitive closures of classes `tc[x]` considered earlier, and indeed there are some connections between the two meanings of transitive. In particular, the transitive closure of the membership relation `E` is

$$\mathtt{trv}[\mathtt{E}]=\mathtt{inverse}[\mathtt{TC}]\circ\mathtt{E}.$$

Iteration can be used to show that `HULL[invar[x]]` is a total function when `x` is thin. When `x` is thin and `y` is a set, the relation `iterate[x,y]` is a set, and hence

$$\mathtt{range}[\mathtt{iterate}[\mathtt{x},\mathtt{y}]]=\mathtt{y}\cup\mathtt{image}[\mathtt{trv}[\mathtt{x}],\mathtt{y}]$$

is a set which contains `y` and is invariant under `x`. In other words, every set is a subset of a set that is invariant under a given thin relation. This fact can be written as follows,

$$\mathtt{thin}[\mathtt{x}]\Rightarrow\mathtt{image}[\mathtt{inverse}[\mathtt{S}],\mathtt{invar}[\mathtt{x}]]=\mathtt{V}.$$

Since the domain of `HULL[x]` is `image[inverse[S], x]`, this completes the proof that `domain[HULL[invar[x]]] = V` whenever `x` is thin. In particular, for the case that `x = inverse[E]` one deduces that `TC = HULL[FULL]` is a total function.

The derivations of many properties of the constructor $\mathtt{trv}$ require an application of iteration. In particular, this was used to derive the principle of well-founded induction. A relation $\mathtt{x}$ is *well-founded* if the only set $\mathtt{y}$ satisfying $\mathtt{y} \subset \mathtt{image}[\mathtt{x},\mathtt{y}]$ is the empty set. For example, the membership relation $\mathtt{E}$ is well-founded if and only if the axiom of regularity holds. Whether or not the axiom of regularity holds, the restriction $\mathtt{id}[\Omega] \circ \mathtt{E}$ of the membership relation to the class Ω of ordinals is well-founded. Another familiar example of a well-founded relation is the restriction $\mathtt{id}[\mathtt{FINITE}] \circ \mathtt{PS}$ of the proper subset relation $\mathtt{PS} = \mathtt{S} \cap \mathtt{Id}'$ to the class of finite sets. Since any subclass of a well-founded relation is well-founded, one can show that $\mathtt{x}$ is well-founded if and only if $\mathtt{P}[\mathtt{x}] \subset \mathtt{WF}$, where $\mathtt{WF}$ is the class of all small well-founded relations. The principle of well-founded induction says that if $\mathtt{x}$ is a well-founded relation whose inverse is thin, then there are no proper classes $\mathtt{y}$ that satisfy $\mathtt{y} \subset \mathtt{image}[\mathtt{x},\mathtt{y}]$. As an application of this, one can show that if $\mathtt{x}$ is a well-founded relation with a thin inverse, then $\mathtt{trv[x]}$ is also well-founded. In particular, since any set is thin, it follows that $\mathtt{WF}$ is invariant under $\mathtt{HULL[TRV]}$.

One of the theorems proved using `Otter` is the principle of $\mathtt{FINITE}$ induction: if the empty set belongs to a class of sets, and if that class is invariant under the cover relation

$$\mathtt{K} = \mathtt{PS} \cap (\mathtt{PS} \circ \mathtt{PS})' = \{\langle \mathtt{x},\mathtt{y}\rangle \mid \exists \mathtt{z}\ (\mathtt{z} \notin \mathtt{x}\ \ \&\ \ \mathtt{y} = \mathtt{x} \cup \{\mathtt{z}\})\},$$

then $\mathtt{FINITE} \subset \mathtt{x}$. Explicitly:

$$\mathtt{0} \in \mathtt{x}\ \ \&\ \ \mathtt{image}[\mathtt{K},\mathtt{x}] \subset \mathtt{x}\ \ \Rightarrow\ \ \mathtt{FINITE} \subset \mathtt{x}.$$

In the course of rederiving this result using the $\mathtt{GOEDEL}$ program, the following general formula for the transitive closure of $\mathtt{K}$ was also derived:

$$\mathtt{Id} \cup \mathtt{trv}[\mathtt{K}] = \mathtt{CUP} \circ \mathtt{id}[\mathtt{V} \times \mathtt{FINITE}] \circ \mathtt{inverse}[\mathtt{FIRST}].$$

Another simple result along these lines is the formula

$$\mathtt{iterate}[\mathtt{K},\{\mathtt{0}\}] = \mathtt{inverse}[\mathtt{CARD}] \circ \mathtt{id}[\omega],$$

for the relation whose vertical sections at the natural numbers are the classes of all sets with a given cardinality. Here $\mathtt{CARD}$ is the cardinality function which assigns to each set the smallest ordinal with which it can be put in one-to-one correspondence, if one exists. Since the axiom of choice is not assumed, the function $\mathtt{CARD}$ need not be total, but its domain does contain the class $\mathtt{FINITE}$.

9 Application to Integer Arithmetic

Because the definitions of $\mathtt{CORE}$ and $\mathtt{HULL}$ functions are not limited to topology, these concepts find important applications in other branches of mathematics. In this section an application to integer addition will be described.

The set $\mathtt{Z}$ of all integers can be defined as the set of equivalence classes of a certain equivalence relation $\mathtt{EQUIDIFF}$ on the set $\omega \times \omega$ of pairs of natural numbers, where pairs $\langle \mathtt{u},\mathtt{v}\rangle$ and $\langle \mathtt{x},\mathtt{y}\rangle$ are considered to be equivalent if the sum of the natural numbers $\mathtt{u}$ and $\mathtt{y}$ equals the sum of $\mathtt{v}$ and $\mathtt{x}$. These equivalence classes are in fact one-to-one functions. For example, the integer zero is the identity function $\mathtt{id}[\omega]$ on the natural numbers, and the integer unity is the successor function on the natural numbers. The non-negative integer `plus[x]` corresponding to the natural number `x` is the function that increments natural numbers by `x`. The negative of the integer `plus[x]` is the function `inverse[plus[x]]`. Each positive integer has domain `omega`, but the domain of `inverse[plus[x]]` is the relative complement of the natural number `x` in ω, that is, the set of all natural numbers greater than or equal to `x`.

The sum of two integers can be defined as the unique integer that contains their composite. As a matter of fact, the composite of two integers is already an integer except for the case that the left factor is positive and the right factor is negative. In that case, the composite is contained in the unique integer obtained by reversing the order of the factors:

$$\mathtt{plus}[\mathtt{x}] \circ \mathtt{inverse}[\mathtt{plus}[\mathtt{y}]] \subset \mathtt{inverse}[\mathtt{plus}[\mathtt{y}]] \circ \mathtt{plus}[\mathtt{x}].$$

This yields the following simple formula for the binary function `INTADD` for integer addition:

$$\mathtt{INTADD} = \mathtt{HULL}[\mathtt{Z}] \circ \mathtt{COMPOSE} \circ \mathtt{id}[\mathtt{Z} \times \mathtt{Z}].$$

From this formula one can derive the familiar properties of integer addition, including the commutative law,

$$\mathtt{INTADD} \circ \mathtt{SWAP} = \mathtt{INTADD},$$

the associative law,

$$\mathtt{INTADD} \circ (\mathtt{Id} \otimes \mathtt{INTADD}) \circ \mathtt{ASSOC} = \mathtt{INTADD} \circ (\mathtt{INTADD} \otimes \mathtt{Id}).$$

The function that takes an integer to its negative is

$$\mathtt{id}[\mathtt{Z}] \circ \mathtt{INVERSE} = \mathtt{INVERSE} \circ \mathtt{id}[\mathtt{Z}].$$

This is an automorphism of integer addition:

$$\mathtt{INVERSE} \circ \mathtt{INTADD} = \mathtt{INTADD} \circ (\mathtt{INVERSE} \otimes \mathtt{INVERSE}).$$

Integer subtraction is expressible in terms of addition and negatives:

$$\mathtt{rotate}[\mathtt{INTADD}] = \mathtt{INTADD} \circ (\mathtt{Id} \otimes \mathtt{INVERSE}).$$

10 Summary

The `CORE` and `HULL` constructors discussed in this paper are useful not only in topology, but have applications in many other branches of mathematics. In group theory, for example, the subgroup generated by a subset of a group is the intersection of all subgroups that contain the given set, a fairly typical application of the hull operation in abstract algebra.

By making available such standard constructors in systems for automated reasoning, and deriving their general properties, a valuable arsenal is created that can be relied upon to furnish the ammunition needed to attack many interesting applications. The lofty dream that automated reasoning and verification systems will one day be used routinely in mathematical reasearch will only be realized if serious efforts are made to connect the abstract principles of automated reasoning with the needs encountered in the everyday practice of modern mathematics, laying a solid foundation upon which one can build the many marvelous edifices that comprise the infrastructure of mathematical research.

References

[1997] Bancerek, G., Closure Operators and Subalgebras, Journal of Formalized Mathematics, vol. 9 (1997), pp. 295–301.

[1999a] Belinfante, J. G. F., Computer proofs in Gödel's class theory with equational definitions for composite and cross, Journal of Automated Reasoning, vol. 22 (1999) pp. 311–339.

[1999b] Belinfante, J. G. F., On computer-assisted proofs in ordinal number theory, Journal of Automated Reasoning, vol. 22 (1999), pp. 341–378.

[2001a] Belinfante, J. G. F., Computer Proofs about Transitive Closure, in International Joint Conference on Automated Reasoning, IJCAR-2001 Short Papers, pp. 11–20, edited by R. Goré, A. Leitsch and T. Nipkow, Technical Report DII 11/01, Siena, Italy, 19–23 June 2001.

[2001b] Belinfante, J. G. F., Discovering Theorems using GOEDEL: A Case Study, in Calculemus-2001, 9th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, 21–22 June 2001, Siena, Italy, edited by Steve Linton and Roberto Sebastiani.

[2003] Belinfante, J. G. F., Reasoning about iteration in Gödel's class theory, in Automated Deduction–CADE-19, Proceedings of the 19th International Conference on Automated Deduction, Miami Beach, FL, USA July–August 2003, edited by F. Baader, Lecture Notes in Artificial Intelligence, vol. 2741, pp. 228–242, Springer Verlag, Berlin, 2003. (ISBN 3-540-40559-3)

[1986] Boyer, R., Lusk, E., McCune, W., Overbeek, R., Stickel M. and Wos, L., Set theory in first orderlogic: clauses for Gödel's axioms, Journal of Automated Reasoning, volume 2 (1986), pages 287–327.

[2004] Chang, K., In Math, Computers Don't Lie. Or Do They?, The New York Times, April 6, 2004.

[1991] Farmer, W. F., and Thayer, F. J., Two Computer-Supported Proofs in Metric-Space Topology, Notices of the American Mathematical Society, vol. 38 (1991), pp. 1133–1138.

[1940] Gödel, K., The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis with the Axioms of Set Theory, Princeton University Press, Princeton, 1940.

[1989] McCune, W. and Wick, C., Automated reasoning about elementary point-set topology, Journal of Automated Reasoning, vol. 5 (1989), pp. 239–255.

[1987] Mendelson, E., Introduction to Mathematical Logic, Third edition, Wadsorth & Brooks/Cole, Monterey, CA, 1987.

[1967] Rubin, J. E., Set Theory for the Mathematician, Holden-Day, San Francisco, 1967.

[1987] Tarski, A., and Givant, S., A Formalization of Set Theory without Variables, American Mathematical Society Colloquium Publications, volume 41, Providence, Rhode Island, 1987.