Reverse Mathematics

Benedict Eastaugh

December 13, 2011

In ordinary mathematical practice, mathematicians prove theorems, reasoning from a fixed[1] set of axioms to a logically derivable conclusion. The axioms in play are usually implicit: mathematicians are not generally in the habit of asserting at the beginning of their papers that they assume ZFC. Given a particular proof, we might reasonably ask which axioms were employed and make explicit the author's assumptions.

Now that we have a set of axioms Γ which are sufficient to prove some theorem φ, we could further inquire whether or not they are *necessary* to prove the theorem, or whether a strictly weaker set of axioms would suffice. To a first approximation, reverse mathematics is the programme of discovering precisely which axioms are both necessary and sufficient to prove any given theorem of ordinary mathematics.

Expanding on this slightly, in reverse mathematics we calibrate the proof-theoretic strength of theorems of ordinary mathematics by proving equivalences between those theorems and systems of axioms in a hierarchy of known strength. This characterisation should provoke at least three immediate questions. Which hierarchy of systems? How do these equivalence proofs work? And just what is "ordinary mathematics"?

An immediate, if not entirely satisfactory answer to the last question is that one synonym for the term is *non-set-theoretic mathematics*. In other words, we mean those parts of mathematics which do not depend on abstract set-theoretical concepts. Typical examples of ordinary mathematics include number theory, calculus, real analysis and computability theory.

The question of how these proofs work is most easily answered by taking a detour through the systems used. To that end, we now turn to the usual way in which reverse mathematics is developed, using *second order arithmetic*.

1 Second order arithmetic and its subsystems

Second order arithmetic is an extension of more familiar systems of arithmetic, such as Peano arithmetic (henceforth PA) and its subsystems. Variables in first order arithmetic range over the natural numbers. Second order arithmetic also has such variables, called *number variables*, but in addition it has *set variables* which range over sets of numbers—in other words, over subsets of the domain of the number variables. They are distinguished by using lowercase letters $i, j, k, \ldots$ for number variables and uppercase letters $X, Y, Z, \ldots$ for set variables. Note that despite the "second order" monicker this is a two-sorted first order language, not a second order one.[2]

The full system of second order arithmetic, also known as Z_2, has three basic kinds of axiom.[3] The first axiomatise the behaviour of the standard numerical vocabulary: the constants 0 and 1; the addition and multiplication functions; and the less-than relation. The second is an induction

[1] For a particular proof.

[2] For more details on the language of second order arithmetic, see Appendix A.

[3] A full listing of these axioms is provided in Appendix B.

axiom, formulated in terms of set membership rather than open formulae. Finally, there is a comprehension scheme,

$$\exists X \forall n(n \in X \leftrightarrow \varphi(n))$$

for any formula φ in which X is not free. It is on restrictions of this comprehension scheme that much of our attention now falls. The following three theories should give something of the flavour of the constellation of different subsystems used in reverse mathematics.[4]

1.1 Recursive comprehension

Perhaps the most fundamental subsystem of Z_2 is RCA_0, named for the *recursive comprehension axiom.* This restricts the comprehension scheme given above to Δ^0_1 formulae, i.e. those which define recursive sets.[5] The 0 subscript in the name indicates, as it does for the other systems we will consider, that RCA_0 only has limited induction.

RCA_0 is often the starting point for proofs in reverse mathematics. It's strong enough to prove the basic results required to get reversals off the ground, yet weak enough that many interesting theorems are out of its reach.

1.2 Arithmetical comprehension

ACA_0 is a much more powerful system than RCA_0. It's named for the *arithmetical comprehension axiom*: the restriction of the full comprehension scheme to arithmetical formulae. An arithmetic formula is one with no set quantifiers, i.e. Σ^0_k for any $k \in \mathbb{N}$, although it may have both set and number parameters.

ACA_0 is the second order counterpart of PA: any formula φ in the first order language of arithmetic (the language in which PA is formulated) which is a theorem of ACA_0 is also a theorem of PA. In fact, an even stronger conservativity result holds: all models of PA can be identified with the first order parts of models of ACA_0, and conversely.

1.3 Weak König's lemma

Unlike the previous two systems, WKL_0 does not offer a distinctive comprehension principle as its defining characteristic. Instead, its axioms are those of RCA_0 plus the principle known as *weak König's lemma.*

Definition 1.1. (Weak König's lemma, WKL) Every infinite subtree of the full binary tree has an infinite path.

Perhaps surprisingly, this statement can actually be formalised in the language of second order arithmetic, using only numbers and sets of numbers, although a few clever coding tricks are required; see Appendix C for details.

[4] A *subsystem of* Z_2 is a formal system S in the language of second order arithmetic where each axiom $\varphi \in S$ is a theorem of Z_2. It should be obvious that this is the case for RCA_0 and ACA_0. Weak König's lemma is a theorem of ACA_0, although WKL_0 is a strictly weaker than ACA_0 and strictly stronger than RCA_0.

[5] A set $A \subseteq \mathbb{N}$ is *recursive* iff its characteristic function $f : \mathbb{N} \to \{0, 1\}$ is total and computable.

2 Proving reversals

Now that we have some sense of the type of theorems we are interested in (those of ordinary mathematics) and the hierarchy of systems which we shall use to investigate them (the subsystems of Z_2), we can examine the details of equivalence proofs in reverse mathematics.

The first step is to formalise our theorem—call it φ—in second order arithmetic. This tends to involve fairly heavy coding, as without the usual panoply of set-theoretic machinery, everything must be expressed in terms of numbers or sets of numbers (see the Appendix for some examples).

Suppose that $\mathsf{ACA}_0 \vdash \varphi$, giving us an upper bound to the strength of the theory required to prove φ. ACA_0 is at least as strong as φ; is it stronger, or is φ actually equivalent to ACA_0?

To answer this question, we add φ as an axiom to the *weak base system* RCA_0 and attempt to derive the axioms of ACA_0. This is the "reverse" part of reverse mathematics, where theorems become axioms and axioms become theorems. The base system is needed because φ will not in general allow us to develop enough of mathematics to carry out the proof.

The axioms of ACA_0 are just those of RCA_0 plus the arithmetical comprehension scheme, so to prove that φ is *equivalent to* ACA_0 *over* RCA_0, we simply show that $\mathsf{RCA}_0 + \varphi \vdash \psi$, where ψ is any instance of arithmetical comprehension.

Note that this is a *theorem scheme*, so the equivalence between the system ACA_0 and the statement φ cannot be formalised at the level of the object language. This does not always hold: if φ were instead equivalent to WKL_0 we could derive the following:

$$\mathsf{RCA}_0 \vdash \mathrm{WKL} \leftrightarrow \varphi.$$

For a more familiar example of a reversal, we turn to the equivalence in set theory between the Axiom of Choice and Zorn's Lemma. Here our base system is ZF set theory, and the equivalence is proved first by assuming AC and deriving Zorn's Lemma, and then by assuming Zorn's Lemma and proving AC. This illustrates the importance of the base theory, as the equivalence actually breaks down in weaker contexts such as that of second order arithmetic.[6]

3 The significance of reversals

Reverse mathematics has historically been motivated by foundational goals: to calibrate the strength of classical mathematical theorems in terms of the axioms required to prove them. The choice of second order arithmetic entails a restriction to essentially countable structures, including those of countable algebra and combinatorics, and separable analysis and topology.

The major subsystems appear to correspond to various restricted foundations of mathematics. RCA_0 is close (with some caveats; see Simpson [2009, remark I.8.9 on p.31]) to Bishop's constructivism. WKL_0 can be seen as a partial realisation of Hilbert's programme of finitistic reductionism; this is explored further in Simpson [1988]. Finally, ACA_0 corresponds to Weyl's predicativism, as developed by Feferman. Simpson [2009, §I.12] details several further correspondences between systems of second order arithmetic and foundational programmes.

With this in mind, classifying the theorems of ordinary mathematics according to which subsystem of second order arithmetic they are equivalent to becomes a powerful tool, since it allows us to develop an understanding of the mathematical consequences of these foundational programmes within a common framework.

Thus far we have discussed these systems largely in terms of axioms, and specifically set existence axioms such as comprehension schemes. An alternative way of formulating the strength

[6]See Dzhafarov and Mummert [2011].

of a particular theorem or system is by appealing to principles of recursion theory. RCA_0 corresponds to the existence of recursive sets and the closure under Turing reducibility and join; WKL_0 to the Jokusch–Soare low basic theorem; and ACA_0 to closure under the Turing jump.

This suggests another way to conceive of reverse mathematics, currently being developed by Shore [2010]. Instead of proof theoretic equivalences, we look at *computable equivalences*, formalising the notion that "harder to prove" means "more difficult to compute".

In a similar vein, Dean and Walsh [2011] offer an account of the significance of reversals in terms of *sameness of computational resource.* They suggest that an equivalence between a particular theorem and axiom system is both *necessary* and *sufficient* for sameness of computational resource. The latter is significant because it allows us to assess whether or not we ought to accept some principle: if we believe that only computable sets exist, then we ought not to accept the completeness theorem, since it reverses to WKL_0. The significance of the former lies in its ability to separate views based on the computational resources which they require: any disagreement about the acceptance of principles should be mirrored by a difference in accepted computational resource.

A The language of second order arithmetic

The language of second order arithmetic, L_2, is a two-sorted first order language, with lowercase *number variables* $i, j, k, \ldots$ ranging over natural numbers and uppercase *set variables* $X, Y, Z, \ldots$ ranging over sets of numbers. *Numerical terms* consist of the constant symbols 0 and 1; number variables; and $t_1 + t_2$ and $t_1 \cdot t_2$ where t_1 and t_2 are numerical terms. *Atomic formulae* consist of $t_1 = t_2$, $t_1 < t_2$ and $t_1 \in X$ where $t1$ and t_2 are numerical terms and X is any set variable. Formulae are built up in the usual way, with distinct number and set quantifiers.

So much for the syntax. As far as the semantics are concerned, a structure for L_2 is one of the form

$$M = \langle |M|, \mathcal{S}_M, +_M, \cdot_M, 0_M, 1_M, <_M \rangle$$

where the number variables range over the domain $|M|$ and set variables range over the set $\mathcal{S}_M \subseteq \mathcal{P}(|M|)$. 0_M and 1_M are constant elements of $|M|$; $+_M$ and $\cdot_M$ are two-place functions on $|M|$; and $<_M$ is a binary relation on $|M|$.

B The axioms of Z_2

The axioms of Z_2 are the universal closures of the following L_2 formulae.

1. $n + 1 \neq 0$
2. $m + 1 = n + 1 \rightarrow m = n$
3. $m + 0 = m$
4. $m + (n + 1) = (m + n) + 1$
5. $m \cdot 0 = 0$
6. $m \cdot (n + 1) = (m \cdot n) + m$
7. $\neg m < 0$

8. $m < n + 1 \leftrightarrow (m < n \vee m = n)$
9. (induction axiom)

$$(0 \in X \wedge \forall n(n \in X \rightarrow n + 1 \in X)) \rightarrow \forall n(n \in X)$$

10. (comprehension scheme)

$$\exists X \forall n(n \in X \leftrightarrow \varphi(n))$$

for all L_2 formulae φ with X not free.

C Formalising weak König's lemma

To demonstrate some of the coding machinery employed in developing mathematics within second order arithmetic, let's consider the statement defined above as *weak König's lemma*, that every subtree of the full binary tree has an infinite path. The full binary tree is $2^{<\mathbb{N}}$, that is, the set of all finite sequences of 0s and 1s. The first step is therefore to develop a way of expressing finite sequences within L_2. We define a *pairing map*

$$(i, j) = (i + j)^2 + i$$

where $k^2 = k \cdot k$. In other words, we can code pairs of numbers as single numbers. RCA_0 allows us to prove the following theorems.

1. $i \leq (i, j)$ and $j \leq (i, j)$.
2. $(i, j) = (i', j') \rightarrow (i = i' \wedge j = j')$.

A *finite sequence* of natural numbers can now be defined as a finite set X such that the following conditions all hold:

1. $\forall n(n \in X \rightarrow \exists i, j(n = (i, j)))$;
2. $\forall i, j, k(((i, j) \in X \wedge (i, k) \in X) \rightarrow j = k)$;
3. $\exists l \forall i(i < l \leftrightarrow \exists j((i, j) \in X)$.

The number l in the third condition is the *length* of X. Finite sets such as X can of course be coded as single numbers,[7] so the set of all (codes of) finite sequences $\mathbb{N}^{<\mathbb{N}}$ is itself just a set of numbers, and in fact it exists by Σ^0_0 comprehension.

A *tree* is a set $T \subseteq \mathbb{N}^{<\mathbb{N}}$ which is closed under initial segments. An *infinite path* is a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $k \in \mathbb{N}$, the initial sequence

$$f[k] = \langle f(0), f(1), \ldots, f(k - 1) \rangle$$

is in T. We can formalise functions as subsets of cartesian products, which are themselves sets of (codes for) pairs as defined above. With this, we have all the ingredients necessary to state weak König's lemma formally in the language of second order arithmetic.

[7]See Simpson [2009, p.67] for details.

References

W. Dean and S. Walsh. Reversals and Sameness of Computational Resource. Talk at the Midwest Phil Math Workshop, Notre Dame, 2011.

D. D. Dzhafarov and C. Mummert. On the strength of the finite intersection principle. *ArXiv e-prints*, Sept. 2011.

R. A. Shore. Reverse Mathematics: The Playground of Logic. *The Bulletin of Symbolic Logic*, Volume 16, Number 3, 2010.

S. G. Simpson. Partial realizations of Hilbert's program. *The Journal of Symbolic Logic*, 53: 349–363, 1988.

S. G. Simpson. *Subsystems of Second Order Arithmetic*. Cambridge University Press, 2nd edition, 2009.