Refutation of recursive comprehension in second-order arithmetic

Abstract: We evaluate 11 basic first-order axioms of which nine are *not* tautologous. Recursive comprehension, as an abstraction of mathematical induction, is derived therefrom in second-order arithmetic and is *not* tautologous. This refutes the use of recursive comprehension in second-order arithmetic. Reverse mathematics relies on recursive comprehension and hence is also refuted.

We assume the method and apparatus of Meth8/VŁ4 with Tautology as the designated *proof* value, **F** as contradiction, N as truthity (non-contingency), and C as falsity (contingency). For results, the 16-valued truth table is row-major and horizontal, or repeating fragments of 128-tables, sometimes with table counts, for more variables. Reproducible transcripts for results are available. (See ersatz-systems.com.)

LET p, q, r, s: A, B, C, D;
~ Not, ¬ ; + Or, ∨, ∪; - Not Or; & And, ∧, ∩; \ Not And;
> Imply, greater than, →, ↦, ≻, ⊃, ⊢, ⊨; < Not Imply, less than, ∈, ≺, ⊂;
= Equivalent, ≡, :=, ⟺, ↔, ≜; @ Not Equivalent, ≠;
% possibility, for one or some, ∃, ◊, M; # necessity, for every or all, ∀, □, L;
(z=z) T as tautology, ⊤, ordinal 3;
(z@z) **F** as contradiction, ∅, Null, ⊥, zero;
(%z<#z) C as contingency, Δ, ordinal 1;
(%z>#z) N as non-contingency, ∇, ordinal 2;
~(y < x) (x ≤ y), (x ⊆y); (A=B) (A~B).

From: en.wikipedia.org/wiki/Second-order_arithmetic

The following axioms are known as the basic axioms.

Axioms governing the successor function and zero:

1. ∀m [S m = 0 → ⊥] ("the successor of a natural number is never zero") (1.1)

(s&#p)>(r@r) ; TTTT TTTT TCTC TCTC (1.2)

2. ∀m ∀n [S m = S n → m = n] ("the successor function is injective") (2.1)

((s&#p)=(s&#q))>(#p=#q) ; TCCT TCCT TTTT TTTT (2.2)

3. ∀n [0 = n ∨ ∃m [S m = n]] ("every natural number is zero or a successor") (3.1)

(((r@r)=#q)&%p)&((s&p)=q) ; CT**FF** CT**FF** C**FF**C C**FF**C (3.2)

Addition defined recursively:

4. ∀m [m + 0 = m] (4.1)

(#p+(r@r))=#p ; TTTT TTTT TTTT TTTT (4.2)

5. $\forall m\ \forall n\ [\ m + S\ n = S\ (\ m + n\ )\ ]$ (5.1)

(#p+(s&q))=(s&(p+q)) ; TCTC TCTC TNTT TNTT (5.2)

Multiplication defined recursively:

6. $\forall m\ [\ m \cdot 0 = 0\ ]$. (6.1)

(#p&(r@r))=(r@r) ; TTTT TTTT TTTT TTTT (6.2)

7. $\forall m\ \forall n\ [\ m \cdot S\ n = (\ m \cdot n\ ) + m\ ]$ (7.1)

(#p&(s&#q))=((p&q)+p) ; T**F**T**F** T**F**T**F** T**F**TN T**F**TN (7.2)

Axioms governing the order relation "<":

8. $\forall m\ [\ m < 0 \rightarrow \bot\ ]$. (“no natural number is smaller than zero”) (8.1)

(#p<(r@r))>(r@r) ; TCTC TCTC TCTC TCTC (8.2)

9. $\forall n\ \forall m\ [\ m < S\ n \leftrightarrow (\ m < n \lor m = n\ )\ ]$ (9.1)

(#p<(s&#q))=((#p<#q)+(#p=#q)) ; **F**NNN **F**NNN **F**NN**F** **F**NN**F** (9.2)

10. $\forall n\ [\ 0 = n \lor 0 < n\ ]$. (“every natural number is zero or bigger than zero”) (10.1)

((r@r)=#q)+((r@r)<#q) ; TTCC TTCC TTCC TTCC (10.2)

11. $\forall m\ \forall n\ [\ (\ S\ m < n \lor S\ m = n\ ) \leftrightarrow m < n\ ]$ (11.1)

(((s&#p)<#q)+((s&#p)=#q))=(#p<#q) ;
FNNN **F**NNN **F**NN**F** **F**NN**F** (11.2)

For the 11 basic axioms of Eqs. 1.2-11.2 as rendered, two as 4.2 and 6.2 are tautologous, and the other nine are *not* tautologous.

From: en.wikipedia.org/wiki/Second-order_arithmetic

Recursive comprehension

[From: en.wikipedia.org/wiki/Reverse_mathematics:
The initials "RCA" stand for "recursive comprehension axiom", where
"recursive" means "computable", as in recursive function.]

The subsystem RCA_0 is ... often used as the base system in reverse mathematics.

It consists of: the basic axioms [Eqs. 1.-11.1 from above] , the Σ^0_1 induction scheme, and the $\Delta^0{}_1$ comprehension scheme. This scheme includes, for every $\Sigma^0{}_1$ formula φ and every $\Pi^0{}_1$ formula ψ, the axiom:

$$\forall m \forall X((\forall n(\varphi(n)\leftrightarrow\psi(n)))\rightarrow\exists Z\forall n(n\in Z\leftrightarrow\varphi(n))) \tag{12.1}$$

LET p, q, r, s, x, z: φ, ψ, m, n, X, Z;

Remark 12.1: For clarity, we distribute the quantifiers to each instance of a variable.

((p&#s)=(q&#s))>((#q<%z)=(p&#s)) ;

```
TTCC TTCC TTTT TTTT(32),
TTTT TTTT TTTC TTTC(32)
```
(12.2)

The formula for recursive comprehension in Eq. 12.2 as rendered is not tautologous. This refutes its use in second-order arithmetic.

We evaluate 11 basic first-order axioms of which nine are *not* tautologous. Recursive comprehension, as an abstraction of mathematical induction, is derived therefrom in second-order arithmetic and is also *not* tautologous. Reverse mathematics requires recursive comprehension and thereby is also refuted.