MODEL THEORY OF ARITHMETIC

Lecture 7: Recursive saturation

Tin Lok Wong

19 November, 2014

> One of the most significant by-products of the study of admissible sets with urelements is the emphasis it has given to recursively saturated models. [... C]ountable recursively saturated models (for finite languages) possess many of the desirable properties of saturated models and special models. The notion of resplendency was introduced to isolate some of these desirable properties.
>
> John Stewart Schlipf [10]

Definition. Fix a *recursive* language $\mathscr{L}$, i.e., one in which all syntactical notions are recursive. Let M be an $\mathscr{L}$-structure and Γ be a class of $\mathscr{L}$-formulas.

(i) A *type* over M is a set of $\mathscr{L}(M)$-formulas $p(\bar{v})$ with finitely many free variables $\bar{v}$ that is consistent with $\mathrm{ElemDiag}(M)$.

(ii) A type $p(\bar{v})$ is *realized* in M if $M \models \bigwedge\!\!\!\bigwedge p(\bar{a})$ for some $\bar{a} \in M$.

(iii) A type $p(\bar{v})$ is *recursive* if it involves only finitely many parameters, say $\bar{c} \in M$, and that

$$\{\ulcorner \theta(\bar{v}, \bar{z}) \urcorner : \theta(\bar{v}, \bar{c}) \in p(\bar{v})\}$$

is recursive.

(iv) The model M is *recursively saturated* if all recursive types over M are realized in M.

(v) A Γ-*type* is a type in which all elements are of the form $\theta(\bar{v}, \bar{c})$ where $\theta \in \Gamma$ and $\bar{c} \in M$.

(vi) The model M is Γ-*recursively saturated* if all recursive Γ-types over M are realized in M.

At first sight, recursive saturation may seem an 'unhappy marriage' between model theory and recursion theory, as some authors put it. Nevertheless, it turns out to work wonderfully well with models of arithmetic.

First and most important of all, since there are only countably many recursive sets, the number of recursive types over any infinite structure M is exactly $\mathrm{card}(M)$. So by an elementary chain argument, every infinite structure for a recursive language has a recursively saturated elementary extension of the same cardinality. In particular, there exist countable recursively saturated models of arithmetic. Notice no countably saturated model of arithmetic is countable because any such model must code all subsets of $\mathbb{N}$.

In Section 7.1, we look at some connections between recursive saturation and arithmetic. In Section 7.2, we show a general model-theoretic property possessed by all countable recursively saturated models called *resplendency*. The two sections are independent of each other.

7.1 Definability of the standard cut

Recursive types are nice because they are always coded in models of sufficiently strong arithmetic. Here we prove a slightly more general proposition that we already saw in Theorem 4.14.

Proposition 7.1 (essentially Scott [11])**.** If $I \subsetneq_{\mathrm{e}} M \models \mathrm{I}\Delta_0 + \exp$, where I is multiplicative, then $(I, \mathrm{Cod}(M/I)) \models \Delta_1^0\text{-CA}$.

Proof. We only show Δ_1-CA. Let $\varphi, \psi \in \Delta_0(I)$ such that

$$I \models \forall x\ \big(\exists u\ \varphi(u,x) \leftrightarrow \forall v\ \psi(v,x)\big). \tag{1}$$

We want $S = \{x \in I : I \models \exists u\ \varphi(u,x)\} \in \mathrm{Cod}(M/I)$. The plan is to overspill (1) into $M \setminus I$, and then reduce S to some Δ_0-definable set in M, which we can code with $\mathrm{I}\Delta_0 + \exp$. Note

$$\begin{aligned} S &= \{x \in I : M \models \varphi(u,x) \text{ for some } u \in I\} \\ &= \{x \in I : M \models \psi(v,x) \text{ for all } v \in I\} \end{aligned}$$

by Proposition 4.12. This implies whenever $b \in I$,

$$M \models \forall u, v, x{<}b\ \big(\varphi(u,x) \to \psi(v,x)\big). \tag{2}$$

Apply Δ_0-overspill to find $b \in M \setminus I$ such that (2) holds. Then for all $x \in I$,

- $M \models \exists u{<}b\ \varphi(u,x) \Rightarrow M \models \forall v{<}b\ \psi(v,x)$ — by (2),
- $M \models \forall v{<}b\ \psi(v,x) \Rightarrow I \models \forall v\ \psi(v,x)$ — since $b > I$ and $\psi \in \Delta_0$,
- $I \models \forall v\ \psi(v,x) \Rightarrow I \models \exists u\ \varphi(u,x)$ — by (1),
- $I \models \exists u\ \varphi(u,x) \Rightarrow M \models \exists u{<}b\ \varphi(u,x)$ — since $b > I$ and $\varphi \in \Delta_0$.

By Δ_0-separation from Theorem 2.7, we know $S^* = \{x < b : M \models \exists u{<}b\ \varphi(u,x)\} \in \mathrm{Cod}(M)$. So $S = S^* \cap I \in \mathrm{Cod}(M/I)$. □

Remark 7.2. Full exp is not needed in this theorem. For the proof above to go through, it suffices to have 2^b for some $b \in M \setminus I$.

One important connection between recursive saturation and arithmetic is about *satisfaction predicates*. First, we need a way to code sequences of bounded lengths. We can do this using the Ackermann interpretation from Lectures 2 and 3. More traditional coding methods involve the Chinese Remainder Theorem or the unique factorization of primes. See Kaye's book [7, Chapter 5] or the Hájek–Pudlák book [6, Section I.1(b)] for the precise definitions.

Notation. Work within $\mathrm{I}\Delta_0 + \exp$. Fix a formula $\mathrm{Seq} \in \Delta_0$ that picks out codes of sequences. We denote by $\mathrm{seqlen}(s)$ the length of the sequence (coded by) s. The ith element in the sequence s is denoted $[s]_i$. The formulas $\ell = \mathrm{seqlen}(s)$ and $x = [s]_i$ are both Δ_0. They satisfy the following.

(1) $\forall s{\in}\mathrm{Seq}\ \exists!\ell\ \mathrm{seqlen}(s) = \ell$.

(2) $\forall s{\in}\mathrm{Seq}\ \forall i{<}\mathrm{seqlen}(s)\ \exists!x\ [s]_i = x$.

(3) $\forall s{\in}\mathrm{Seq}\ \forall i{\geqslant}\mathrm{seqlen}(s)\ \forall x\ [s]_i \neq x$.

(4) $\forall i, s, x\ \big(x = [s]_i \to x \leqslant s \wedge i \leqslant s\big)$.

(5) $\forall s, t{\in}\mathrm{Seq}\ \big(\mathrm{seqlen}(s) \leqslant \mathrm{seqlen}(t) \wedge \forall i{<}\mathrm{seqlen}(s)\ [s]_i \leqslant [t]_i \to s \leqslant t\big)$.

(6) $\exists s{\in}\mathrm{Seq}\ \mathrm{seqlen}(s) = 0$.

(7) $\forall s{\in}\mathrm{Seq}\ \forall x\ \exists s'{\in}\mathrm{Seq}\ \big(\mathrm{seqlen}(s') = \mathrm{seqlen}(s) + 1 \wedge \forall i{<}\mathrm{seqlen}(s)\ [s']_i = [s]_i \wedge [s']_{\mathrm{seqlen}(s)} = x\big)$.

We write $[\bar{x}]$ for the code of the sequence $\bar{x}$.

A satisfaction predicate is a formula $S(\theta, [\bar{x}])$ that can evaluate the truth value of a given formula θ under a particular variable assignment $[\bar{x}]$. In other words, such $S(\theta, [\bar{x}])$ behaves the same as $\theta(\bar{x})$. Formally, one needs a lot of care when defining satisfaction, e.g., to make sure formulas and variable assignments match properly. We are content with an informal approach here, which should pose no risk of ambiguity in our context. We should, however, be aware of the complications, and convince ourselves that the missing details can in theory be filled in.

Theorem 7.3 (essentially Kleene [8]). There is a formula $\Delta_0\text{-Sat}(\theta, s)$ that is Δ_1 over $\mathrm{I}\Delta_0 + \exp$ for which $\mathrm{I}\Delta_0 + \exp$ proves

(a) $\forall \theta, s\ \big(\Delta_0\text{-Sat}(\theta, s) \to \theta \in \Delta_0 \wedge s \in \mathrm{Seq}\big)$; and

(b) Tarski's clauses for satisfaction for Δ_0-formulas.

In particular, for every $\theta \in \Delta_0$,

$$\mathrm{I}\Delta_0 + \exp \vdash \forall \bar{x}\ \big(\theta(\bar{x}) \leftrightarrow \Delta_0\text{-Sat}(\theta, [\bar{x}])\big).$$

Proof sketch. Let U be a *universal program*, i.e., a program for which

- P halts on input $\bar{x}$ if and only if U halts on input $\ulcorner P \urcorner, [\bar{x}]$; and
- if P halts on input $\bar{x}$, then $U(\ulcorner P \urcorner, [\bar{x}]) = P(\bar{x})$

for all programs P and all inputs $\bar{x}$. Using Fact 1.6, find $\upsilon \in \Sigma_1$ such that $U = \mathrm{Prog}\langle \upsilon \rangle$. Define

$$\Delta_0\text{-Sat}(\theta, [\bar{x}]) = \upsilon(\ulcorner \mathrm{Prog}\langle \theta \rangle \urcorner, [\bar{x}]).$$

It satisfies Tarski's clauses and is equivalent to $\neg\Delta_0\text{-Sat}(\neg\theta, [\bar{x}])$, because Δ_0-programs always halt. So $\Delta_0\text{-Sat} \in \Delta_1$. All these can be formalized in $\mathrm{I}\Delta_0 + \exp$. See Kaye's book [7, Chapter 9] or the Hájek–Pudlák book [6, Subsection I.1(d)] for the details. ☐

Using this, one can define satisfaction for formulas higher in the arithmetic hierarchy.

Definition. Fix a formula Δ_0-Sat as in Theorem 7.3. Define $\Sigma_0\text{-Sat} = \Pi_0\text{-Sat} = \Delta_0\text{-Sat}$. Modulo some syntactical checks and transformations, define for every $n \in \mathbb{N}$,

(i) $\Sigma_{n+1}\text{-Sat}\big(\exists \bar{y}\ \eta(\bar{x}, \bar{y}), [\bar{x}]\big) = \exists [\bar{y}]\ \Pi_n\text{-Sat}(\eta, [\bar{x}, \bar{y}])$; and

(ii) $\Pi_{n+1}\text{-Sat}(\theta, [\bar{x}]) = \neg\Sigma_{n+1}\text{-Sat}(\neg\theta, [\bar{x}])$.

Observations. (1) $\Sigma_{n+1}\text{-Sat} \in \Sigma_{n+1}$ and $\Pi_{n+1}\text{-Sat} \in \Pi_{n+1}$ over $\mathrm{I}\Delta_0 + \exp$ for every $n \in \mathbb{N}$.

(2) If Γ is Σ_n or Π_n for some $n \in \mathbb{N}$, then for all $\theta \in \Gamma$,

$$\mathrm{I}\Delta_0 + \exp \vdash \forall \bar{x}\ \big(\theta(\bar{x}) \leftrightarrow \Gamma\text{-Sat}(\theta, [\bar{x}])\big).$$

A simple application of these satisfaction predicates is a characterization of Σ_{n+1}-recursive saturation in terms of the definability of the standard cut, which, in a sense, says

$$\text{recursive saturation} = \text{overspill at } \mathbb{N}.$$

Theorem 7.4 (H. Friedman [5], implicitly). For all $n \in \mathbb{N}$ and all $M \models \mathrm{B}\Sigma_{n+1} + \exp$, the following are equivalent.

(a) M is Σ_{n+1}-recursively saturated.

(b) $\mathbb{N}$ is not Σ_{n+1}-definable in M.

Proof. Let us first prove the easier direction (a) $\Rightarrow$ (b). Suppose (b) does not hold. Let $\theta \in \Sigma_{n+1}(M)$ that defines $\mathbb{N}$ in M. Then the recursive Σ_{n+1}-type

$$p(v) = \{\theta(v)\} \cup \{v > n : n \in \mathbb{N}\}$$

is not realized in M. So (a) fails.

Consider the implication (b) $\Rightarrow$ (a). Suppose (b) holds. Notice $M \neq \mathbb{N}$ as a result. Let $p(\bar{v})$ be a recursive Σ_{n+1}-type over M. The plan is to use (b) to overspill the finite satisfiability of p to full satisfiability in M. To execute the plan, recall $(\mathbb{N}, \mathrm{Cod}(M/\mathbb{N})) \models \Delta^0_1\text{-CA}$ from Proposition 7.1. Find $c \in M$ that codes p. Since p is a type over M, for every $k \in \mathbb{N}$,

$$M \models \underbrace{\exists \bar{v}\ \forall \theta{<}k\ \big(\underbrace{\theta \in \mathrm{Ack}(c)}_{\Delta_0} \to \underbrace{\Sigma_{n+1}\text{-Sat}(\theta, [\bar{v}])}_{\Sigma_{n+1}}\big)}_{\Sigma_{n+1} \text{ over } \mathrm{B}\Sigma_{n+1}}.$$

This overspills into $M \setminus \mathbb{N}$ by (b). Any witness to the overspilled statement realizes p. So (a) holds. ☐

It then follows from Proposition 3.3 that all nonstandard models of $\mathrm{I}\Sigma_{n+1}$, where $n \in \mathbb{N}$, are Σ_{n+1}-recursively saturated. In particular, all nonstandard models of PA are Σ_n-recursively saturated for every $n \in \mathbb{N}$. However, we will meet in Lecture 9 some nonstandard models of PA that are not recursively saturated. One can also find nonstandard models of $\mathrm{B}\Sigma_0 + \exp$ (in which $\mathbb{N}$ must not be Σ_0-definable) that are not Σ_0-recursively saturated [7, Theorem 10.10].

Remark 7.5. The argument in the proof of Theorem 7.4 also shows that all models constructed by means of the Arithmetized Completeness Theorem within a nonstandard model of RCA_0 are recursively saturated.

7.2 Resplendency

Let us leave arithmetic for the moment, and study some general model theory. If one has not come across the word *resplendent* before, then she/he is invited find out its literal meaning in a dictionary for amusement.

Definition. A structure M in a recursive language $\mathscr{L}$ is *resplendent* if

> whenever φ is a formula in a recursive language $\mathscr{L}^* \supseteq \mathscr{L}$ and $\bar{c} \in M$ such that $\varphi(\bar{c})$ is consistent with $\mathrm{ElemDiag}(M)$, there is an expansion of M satisfying $\varphi(\bar{c})$.

Informally speaking, a structure is resplendent if and only if it has so many types of subsets that it can expand to model whatever sentence it can consistently model in an expansion. Compare this with saturation: a structure is saturated if and only if it has so many types of elements that it possesses whatever kind of elements it can consistently possess. These two notions of richness turn out to coincide for arithmetic as well as in many other circumstances.

Theorem 7.6 (Barwise, Ressayre, independently)**.** Let M be a countable recursively saturated structure for a recursive language $\mathscr{L}$. Then M is resplendent. In fact, if $\mathscr{L}^*$ is a recursive extension of $\mathscr{L}$, and Φ is a recursive set of $\mathscr{L}^*(M)$-formulas consistent with $\mathrm{ElemDiag}(M)$ that involves only finitely many parameters from M, then M expands to a model of Φ.

Barwise's proof [2, Theorem IV.5.7] uses admissible sets. We follow the more elementary proof from Ressayre [9, Theorem 2.3] here. It will be handy to be able to realize r.e. types in recursively saturated structures.

Craig's Trick [4]. Every r.e. theory in a recursive language is equivalent to a recursive theory.

Proof. Let T be a recursive theory in a recursive language $\mathscr{L}$. Use Corollary 1.7 to find $\theta \in \Delta_0$ such that $T = \{\sigma \in \mathscr{L} : \mathbb{N} \models \exists x\ \theta(x, \ulcorner\sigma\urcorner)\}$. Then T is equivalent to the recursive theory

$$\big\{\underbrace{\sigma \wedge \sigma \wedge \cdots \wedge \sigma}_{(n+1)\text{-many } \sigma\text{'s}} \in \mathscr{L} : n \in \mathbb{N} \models \exists x{<}n\ \theta(x, \ulcorner\sigma\urcorner)\big\}. \qquad \square$$

Proof of Theorem 7.6. We will find by recursion sentences $\varphi_0, \varphi_1, \ldots \in \mathscr{L}^*(M)$ such that

$$\Phi_\omega = \Phi + \{\varphi_n : n \in \mathbb{N}\}$$

is complete, consistent, Henkinized, and includes $\mathrm{ElemDiag}(M)$. The term model of Φ_ω will then be an expansion of M satisfying Φ. Recursive saturation will provide us enough constant symbols in $\mathscr{L}^*(M)$ to Henkinize without further expanding the language.

At each step $n \in \mathbb{N}$, we inductively assume

$$\Phi_n = \Phi + \{\varphi_k : k < n\} \text{ is consistent with } \mathrm{ElemDiag}(M). \qquad (*)$$

By hypothesis, we know Φ_0 satisfies this condition. Now suppose Φ_n is already found, and it satisfies $(*)$. Consider the $\mathscr{L}^*(M)$-formula $\psi_n(v)$, which comes from some fixed enumeration of all $\mathscr{L}^*(M)$-formulas. The countability of M is used here to find such an enumeration. By insisting φ_n to be either $\forall v\ \neg\psi_n(v)$ or $\psi_n(a)$ for some $a \in M$, we force a Henkin axiom for ψ_n to hold in Φ_ω. This, in particular, makes Φ_ω complete too.

Without loss of generality, suppose we cannot define φ_n to be $\forall v\ \neg\psi_n(v)$ while maintaining the inductive condition. Let $c_1, c_2, \ldots, c_\ell \in M$ be the finitely many parameters that appear in Φ_n, and $d_1, d_2, \ldots, d_m \in M$ be the parameters that appear in ψ_n but not in Φ_n. We will indicate all parameters from M for the rest of the proof. In particular, we write $\Phi_n = \Phi_n(\bar{c})$ and $\psi_n(v) = \psi(v, \bar{c}, \bar{d})$. Find $\alpha(\bar{c}, \bar{d}) \in \mathrm{ElemDiag}(M)$ such that $\Phi_n(\bar{c}) + \alpha(\bar{c}, \bar{d}) \vdash \exists v\ \psi_n(v, \bar{c}, \bar{d})$. We may assume $\alpha(\bar{c}, \bar{d})$ contains no parameters other than those already indicated because any further parameters can be quantified out existentially.

Without loss of generality, we suppose further that setting φ_n to be $\psi_n(c_i, \bar{c}, \bar{d})$ for any $i = 1, 2, \ldots, \ell$ would violate the inductive condition. For each such i, fix $\beta_i(\bar{c}, \bar{d}) \in \mathrm{ElemDiag}(M)$ such that $\Phi_n(\bar{c}) + \beta_i(\bar{c}, \bar{d}) \vdash \neg\psi_n(c_i, \bar{c}, \bar{d})$. As in the previous paragraph, we may assume no further parameters appear here.

Recall from Section 4.2 that provability in a recursive theory is r.e. So the set

$$p(v) = \left\{\theta(v, \bar{c}, \bar{d}) \in \mathscr{L}(\bar{c}, \bar{d}) : \Phi_n(\bar{c}) \vdash \forall v, \bar{z}\ \big(\psi_n(v, \bar{c}, \bar{z}) \wedge v \notin \{\bar{c}\} \to \theta(v, \bar{c}, \bar{z})\big)\right\}$$

is r.e. By Craig's Trick, recursive saturation applies to p. To show that p is a type over M, it suffices to prove the consistency of single formulas from p with $\mathrm{ElemDiag}(M)$, because the conjunction of finitely many formulas from p is again an element of p. Now, if $\theta(v, \bar{c}, \bar{d}) \in p(v)$, then

$$\begin{array}{lll}
& \Phi_n(\bar{c}) \vdash \forall v, \bar{z}\ \big(\psi_n(v, \bar{c}, \bar{z}) \wedge v \notin \{\bar{c}\} \to \theta(v, \bar{c}, \bar{z})\big) & \text{by the definition of } p, \\
\therefore & \Phi_n(\bar{c}) \vdash \forall \bar{z}\ \Big(\alpha(\bar{c}, \bar{z}) \wedge \bigwedge_{i=1}^{\ell} \beta_i(\bar{c}, \bar{z}) \to \exists v\ \theta(v, \bar{c}, \bar{z})\Big) & \text{by the choices of } \alpha(\bar{c}, \bar{d}), \beta_i(\bar{c}, \bar{d}), \\
\therefore & M \models \forall \bar{z}\ \Big(\alpha(\bar{c}, \bar{z}) \wedge \bigwedge_{i=1}^{\ell} \beta_i(\bar{c}, \bar{z}) \to \exists v\ \theta(v, \bar{c}, \bar{z})\Big) & \text{by } (*)\text{, noting the formula is in } \mathscr{L}(M), \\
\therefore & M \models \exists v\ \theta(v, \bar{c}, \bar{d}) & \text{because } M \models \alpha(\bar{c}, \bar{d}) \wedge \bigwedge_{i=1}^{\ell} \beta_i(\bar{c}, \bar{d}).
\end{array}$$

Apply recursive saturation to find $a \in M$ that realizes p. Notice $a \notin \{\bar{c}\}$ because the formula $v \neq c_i$ is in $p(v)$ for every $i \in \{1, 2, \ldots, \ell\}$.

Set $\varphi_n = \psi_n(a, \bar{c}, \bar{d})$. Then the inductive condition again holds because if $\theta(a, \bar{c}, \bar{d}) \in \mathscr{L}(a, \bar{c}, \bar{d})$ such that $\Phi_n(\bar{c}) + \psi_n(a, \bar{c}, \bar{d}) \vdash \theta(a, \bar{c}, \bar{d})$, then

$$\begin{array}{lll}
& \Phi_n(\bar{c}) \vdash \forall v, \bar{z}\ \big(\psi_n(v, \bar{c}, \bar{z}) \to \theta(v, \bar{c}, \bar{z})\big) & \text{since } \Phi(\bar{c}) \text{ does not involve } a, \bar{d}, \\
\therefore & \theta(v, \bar{c}, \bar{d}) \in p(v) & \text{by the definition of } p, \\
\therefore & M \models \theta(a, \bar{c}, \bar{d}) & \text{since } a \text{ realizes } p.
\end{array}$$

Thus we get $\Phi_{n+1}(a, \bar{c}, \bar{d})$ as required. □

Further exercises

These exercises explore the relationship between recursive saturation and *partial (nonstandard) inductive satisfaction classes*. We work in a fixed nonstandard $M \models \mathrm{PA}$ throughout.

Definition. A *partial inductive satisfaction class* on M is a subset $S \subseteq M^2$ that satisfies

(i) *induction with* M, i.e., the structure (M, S), where S is interpreted as a new predicate, satisfies the induction scheme for all formulas in the expanded language $\mathscr{L}_{\mathrm{A}} \cup \{S\}$; and

(ii) Tarski's clauses for satisfaction for all standard $\mathscr{L}_{\mathrm{A}}$-formulas.

In particular, condition (ii) above implies that for all formulas $\theta \in \mathscr{L}_{\mathrm{A}}$ and all $\bar{x} \in M$,

$$M \models \theta(\bar{x}) \quad \Leftrightarrow \quad S(\ulcorner\theta\urcorner, [\bar{x}]).$$

(a) Review the proof of Tarski's theorem on the undefinability of truth. Show that no partial (inductive) satisfaction class on M is definable in M.

(b) Imitate the proof of Theorem 7.4, or otherwise, to show that if M has a partial inductive satisfaction class S, then M is recursively saturated.

(c) Use resplendency to show that if M is countable and recursively saturated, then it has a partial inductive satisfaction class.

Further comments

The arithmetic hierarchy is strict

A diagonalization argument similar to that for Tarski's theorem on the undefinability of truth shows Δ_0-Sat $\notin \Delta_0$, and for every $n \in \mathbb{N}$,

$$\Sigma_{n+1}\text{-Sat} \notin \Pi_{n+1} \quad \text{and} \quad \Pi_{n+1}\text{-Sat} \notin \Sigma_{n+1}.$$

Therefore, the arithmetic hierarchy does not collapse, cf. the Further exercises in Lecture 1.

Truth definition without exponentiation

The existence of satisfaction predicates without exp turns out to relate to Question 3.9. See the paper by Adamowicz, Kołodziejczyk, and Paris [1] for the most recent status of the problems.

Resplendency implies recursive saturation

The stronger version of Theorem 7.6 clearly admits a converse. There is also a partial converse to the weaker version. In particular, a countable $\mathscr{L}_\text{A}$-structure is recursively saturated if and only if it is resplendent. Uncountable recursively saturated structures need not be resplendent [7, page 251].

Theorem 7.7 (Jon Barwise [10]). Let $\mathscr{L}$ be a language with only finitely many non-logical symbols. Then every resplendent structure for $\mathscr{L}$ is recursively saturated.

References

[1] Zofia Adamowicz, Leszek Aleksander Kołodzieczyk, and Jeff B. Paris. Truth definitions without exponentiation and the Σ_1 collection scheme. *The Journal of Symbolic Logic*, 77(2):649–655, June 2012.

[2] Jon Barwise. *Admissible Sets and Structures: An Approach to Definability Theory.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1975.

[3] Jon Barwise and John Stewart Schlipf. An introduction to recursively saturated models and resplendent models. *The Journal of Symbolic Logic*, 41(2):531–536, June 1976.

[4] William Craig. On axiomatizability within a system. *The Journal of Symbolic Logic*, 18(1):30–32, March 1953.

[5] Harvey M. Friedman. Countable models of set theory. In Adrian R.D. Mathias and Hartley Rogers, editors. *Cambridge Summer School in Mathematical Logic*, volume 337 of *Lecture Notes in Mathematics*, pages 539–573. Springer-Verlag, Berlin, 1973.

[6] Petr Hájek and Pavel Pudlák. *Metamathematics of First-Order Arithmetic.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1993.

[7] Richard Kaye. *Models of Peano Arithmetic*, volume 15 of *Oxford Logic Guides.* Clarendon Press, Oxford, 1991.

[8] Stephen Cole Kleene. Recursive predicates and quantifiers. *Transations of the American Mathematical Society*, 53(1):41–73, January 1943.

[9] Jean-Pierre Ressayre. Models with compactness properties relative to an admissible language. *Annals of Mathematical Logic*, 11(1):31–55, May 1977.

[10] John Stewart Schlipf. Toward model theory through recursive saturation. *The Journal of Symbolic Logic*, 43(2):183–206, June 1978.

[11] Dana Scott. Algebras of sets binumerable in complete extensions of arithmetic. In Jacob C.E. Dekker, editor. *Recursive Function Theory*, volume V of *Proceedings of Symposia in Pure Mathematics*, pages 117–121. American Mathematical Society, Providence, Rhode Island, 1962.