Variations on resplendency and recursive saturation

Fredrik Engström

These slides are available at:

http://engstrom.morot.org/
Preliminaries

- All languages will be recursive extensions of the language of arithmetic:

\[ L_A = \{ +, \cdot, 0, 1, < \} . \]
All languages will be recursive extensions of the language of arithmetic:

\[ \mathcal{L}_A = \{ +, \cdot, 0, 1, < \} . \]

All models will be models of PA\(^*\), i.e., PA together with induction axioms for the full language.
Recursive saturation...
A *type* $p(x, a)$ over a model $M$ is a set of formulas with parameter $a \in M$, such that there is an elementary extension $N$ of $M$ and element $n \in N$ satisfying $N \models p(n, a)$. 
Recursive saturation...

- A *type* $p(x, a)$ over a model $M$ is a set of formulas with parameter $a \in M$, such that there is an elementary extension $N$ of $M$ and element $n \in N$ satisfying $N \models p(n, a)$.

- $M$ is *recursively saturated* if all recursive types over $M$ are realized.
Recursive saturation...

- A type \( p(x, a) \) over a model \( M \) is a set of formulas with parameter \( a \in M \), such that there is an elementary extension \( N \) of \( M \) and element \( n \in N \) satisfying \( N \models p(n, a) \).

- \( M \) is recursively saturated if all recursive types over \( M \) are realized.

- Any model \( M \) has an elementary extension of the same cardinality which is recursively saturated.
SSy$(M) \subseteq \mathcal{P}(\omega)$ is the standard system of $M$, i.e., the collection of standard parts of parameter definable sets; i.e., the collection of all sets of the form $\{ n \in \omega \mid M \models \varphi(n, a) \}$, where $a \in M$. 
SSy($M$) $\subseteq \mathcal{P}(\omega)$ is the standard system of $M$, i.e., the collection of standard parts of parameter definable sets; i.e., the collection of all sets of the form \[ \{ n \in \omega \mid M \models \varphi(n, a) \} , \] where $a \in M$.

For any $M$, all recursive sets are in SSy($M$).
A digression
Stronger versions of saturation
Stronger versions of saturation

There are stronger variants of recursive saturation:
Stronger versions of saturation

There are stronger variants of recursive saturation:

- Arithmetic saturation: rec sat plus $SS_y(M)$ closed under the jump operator.
Stronger versions of saturation

There are stronger variants of recursive saturation:

- Arithmetic saturation: rec sat plus $SSy(M)$ closed under the jump operator.

- $\beta$-saturation: rec sat plus $SSy(M)$ is a $\beta$-model, i.e., for every $\Sigma^1_1$-formula $\Theta(X)$ and $A \in SSy(M)$; if $\mathbb{N}_2 \models \Theta(A)$ then $SSy(M) \models \Theta(A)$.

$\mathbb{N}_2$ is the standard model of second-order arithmetic.
Arithmetic saturation
Arithmetic saturation

First introduced by Kaye, Kossak and Kotlarski when they proved that a countable recursively saturated model of arithmetic has a maximal automorphism iff the model is arithmetically saturated.
A maximal automorphism is an automorphism moving all non-definable points.
$\beta$-saturation...
\(\beta\)-saturation...

- Jonathan Stavi, in the -80, (almost) proved that a short cofinally expandable model is \(\beta\)-saturated.
Jonathan Stavi, in the -80, (almost) proved that a short cofinally expandable model is $\beta$-saturated. However; Solovay later proved that no short cofinally expandable models exist.
There is a variant of resplendency (an expandability property) which is equivalent to $\beta$-saturation:
There is a variant of resplendency (an expandability property) which is equivalent to $\beta$-saturation:

For every $T, p(x, a) \in SSy(M)$, where $T$ is a theory and $p(x)$ is a type, both in a recursive extension of the language of $(M, a)$, $a \in M$:

If $Th(M, a) + T + p^\uparrow$ has a model then there is an elementary submodel $a \in L$ of $M$ with an expansion satisfying $T + p^\uparrow$.

$p^\uparrow$ means that $p(x, a)$ is omitted.
Back from the digression
The standard predicate
The standard predicate

- The standard predicate, $st$, is the predicate of standard numbers.
The standard predicate

- The standard predicate, \( \text{st} \), is the predicate of standard numbers.

- No model \((M, \text{st})\) is recursively saturated since the type

\[
\{ \, x > n \land \text{st}(x) \mid n \in \omega \, \}
\]

is omitted.
Standard recursive saturation
Standard recursive saturation

- A *standard type over* \( M \) is a type over \((M, \text{st})\) such that there is an \( \omega \)-saturated elementary extension of \( M \) realizing the type.
Standard recursive saturation

- A *standard type over* $\mathcal{M}$ is a type over $(\mathcal{M}, \text{st})$ such that there is an $\omega$-saturated elementary extension of $\mathcal{M}$ realizing the type.

- A model is *standard recursively saturated* (std rec sat) if all recursive standard types are realized.
A *standard type over $M$* is a type over $(M, st)$ such that there is an $\omega$-saturated elementary extension of $M$ realizing the type.

A model is *standard recursively saturated* (std rec sat) if all recursive standard types are realized.

Any type over $M$ (in which $st$ does not occur) is a standard type.
An equivalence
An equivalence

A countable recursively saturated model is std rec sat iff
An equivalence

A countable recursively saturated model is std rec sat iff

for all standard types $p(x, a) \in \text{SSy}(M)$ over $(M, st)$ there is a complete standard type $q(x, a) \in \text{SSy}(M)$ extending $p(x, a)$. 
The proof of the equivalence
The proof of the equivalence

Lemma: If $M$ is countable and std rec sat, and $M < N$ is $\omega$-saturated then $(M, \text{st}) < (N, \text{st})$. 
The proof of the equivalence

- Lemma: If $M$ is countable and std rec sat, and $M \prec N$ is $\omega$-saturated then $(M, \text{st}) \prec (N, \text{st})$.

- Thus, any type $\text{tp}_{(M, \text{st})}(m/a)$, where $M$ is std rec sat, is a standard type.
The proof of the equivalence

Lemma: If $M$ is countable and std rec sat, and $M < N$ is $\omega$-saturated then $(M, \text{st}) < (N, \text{st})$.

Thus, any type $tp_{(M,\text{st})}(m/a)$, where $M$ is std rec sat, is a standard type.

$\Rightarrow$ Let $M$ be std rec sat, and $p(x, a) \in \text{SSy}(M)$ a std type. Let $m \in M$ realize $p(x, a)$. Then, $p(x, a) \subseteq tp_{(M,\text{st})}(m/a) \in \text{SSy}(M)$.
The proof of the equivalence

Lemma: If $M$ is countable and std rec sat, and $M \prec N$ is $\omega$-saturated then $(M, st) \prec (N, st)$.

Thus, any type $tp_{(M, st)}(m/a)$, where $M$ is std rec sat, is a standard type.

$\Rightarrow$ Let $M$ be std rec sat, and $p(x, a) \in SSy(M)$ a std type. Let $m \in M$ realize $p(x, a)$. Then, $p(x, a) \subseteq tp_{(M, st)}(m/a) \in SSy(M)$.

$\Leftarrow$ By a Henkin type construction.
The standard system...
The standard system...

Let $M$ be a std rec sat model.
Let $M$ be a std rec sat model. Then

(1) $SSy(M)$ is a $\beta_\omega$-model of second-order arithmetic, i.e., as second order models $SSy(M) \prec \mathbb{N}_2$, where $\mathbb{N}_2$ is the standard second-order model of arithmetic.
The standard system...

Let $M$ be a std rec sat model. Then

(1) $\text{SSy}(M)$ is a $\beta_\omega$-model of second-order arithmetic, i.e., as second order models $\text{SSy}(M) \prec \mathbb{N}_2$, where $\mathbb{N}_2$ is the standard second-order model of arithmetic.

(2) $\text{SSy}(M)$ is closed under the following operation:

$$A \subseteq \omega \mapsto \text{Th}(\mathbb{N}_2, A).$$
Under certain set-theoretic assumptions ($V = L$ or projective determinacy) we have $(2) \Rightarrow (1)$. 
Under certain set-theoretic assumptions ($V = L$ or projective determinacy) we have $(2) \implies (1)$.

Question: Are conditions (1) and (2) also sufficient, i.e., is any countable recursively saturated model satisfying condition (1) and (2) std rec sat?
The end

That’s all folks!