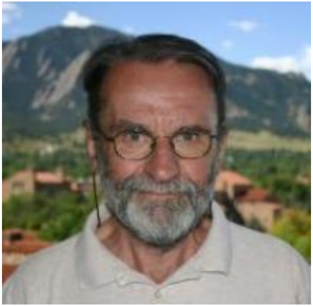
Omitting Types



“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

Omitting Types Theorem. (A. Ehrenfeucht.) Let T be a satisfiable theory in a countable language L. If Φ is a countable set of unsupported partial types over T. There is a countable model of T that omits all types in Φ.

“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

Omitting Types Theorem. (A. Ehrenfeucht.) Let T be a satisfiable theory in a countable language L. If Φ is a countable set of unsupported partial types over T. There is a countable model of T that omits all types in Φ.

Idea: Copy Henkin’s proof of the Completeness Theorem to extend T to a complete theory T_∞ with witnesses.

“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

Omitting Types Theorem. (A. Ehrenfeucht.) Let T be a satisfiable theory in a countable language L. If Φ is a countable set of unsupported partial types over T. There is a countable model of T that omits all types in Φ.

Idea: Copy Henkin's proof of the Completeness Theorem to extend T to a complete theory T_∞ with witnesses.

As the construction progresses, for each $p \in \Phi$,

“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

Omitting Types Theorem. (A. Ehrenfeucht.) Let T be a satisfiable theory in a countable language L. If Φ is a countable set of unsupported partial types over T. There is a countable model of T that omits all types in Φ.

Idea: Copy Henkin's proof of the Completeness Theorem to extend T to a complete theory T_∞ with witnesses.

As the construction progresses, for each $p \in \Phi$, ensure that for every tuple $\mathbf{c}$ of constants, $\neg\varphi(\mathbf{c})$ is added to T_∞ for some formula with $\varphi(\mathbf{x}) \in p$.

“Any fool can realize a type, but it takes a model theorist to omit one.” –from Saturated Model Theory, G. Sacks

Omitting Types Theorem. (A. Ehrenfeucht.) Let T be a satisfiable theory in a countable language L. If Φ is a countable set of unsupported partial types over T. There is a countable model of T that omits all types in Φ.

Idea: Copy Henkin's proof of the Completeness Theorem to extend T to a complete theory T_∞ with witnesses.

As the construction progresses, for each $p \in \Phi$, ensure that for every tuple $\mathbf{c}$ of constants, $\neg\varphi(\mathbf{c})$ is added to T_∞ for some formula with $\varphi(\mathbf{x}) \in p$.

The Henkin model cannot realize any type in Φ, because we forced that.

Organization of proof for $\Phi = \{p\}$ a single n-type

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$.
   Goals:
   1. $T \cup \{\theta_i \mid i \in \omega\}$ is a Henkin L_∞-theory.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$. Goals:
    1. $T \cup \{\theta_i \mid i \in \omega\}$ is a Henkin L_∞-theory.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$. Goals:
    1. $T \cup \{\theta_i \mid i \in \omega\}$ is a Henkin L_∞-theory.
    2. The Henkin model omits p.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$. Goals:
    1. $T \cup \{\theta_i \mid i \in \omega\}$ is a Henkin L_∞-theory.
    2. The Henkin model omits p.

Organization of proof for $\Phi = \{p\}$ a single n-type

1. Let L_∞ be the language obtained from L by adding countably many new constant symbols.
2. Enumerate with ω all L_∞-sentences: $\sigma_0, \sigma_1, \ldots$.
3. Enumerate with ω all n-tuples of constant symbols $\mathbf{c}_0, \mathbf{c}_1, \ldots$.
4. Construct a sequence of increasingly stronger L_∞-sentences $\theta_0, \theta_1, \ldots$. Goals:
    1. $T \cup \{\theta_i \mid i \in \omega\}$ is a Henkin L_∞-theory.
    2. The Henkin model omits p.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$,

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$,

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

Want T_∞ to
(1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k+1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i+1 = 3k+1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k+2$: ensuring witnesses)

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k + 2$: ensuring witnesses)

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k+1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i+1 = 3k+1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k+2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k+1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i+1 = 3k+1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k+2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.
   Choose a constant c that does not appear in $T \cup \{\theta_i\}$.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k + 2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.
   Choose a constant c that does not appear in $T \cup \{\theta_i\}$.
   Let $\theta_{i+1} = \theta_i \wedge \varphi(c)$.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k+1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i+1 = 3k+1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k+2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.
   Choose a constant c that does not appear in $T \cup \{\theta_i\}$.
   Let $\theta_{i+1} = \theta_i \wedge \varphi(c)$.
   If $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is not consistent, do nothing.

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k+1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i+1 = 3k+1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k+2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.
   Choose a constant c that does not appear in $T \cup \{\theta_i\}$.
   Let $\theta_{i+1} = \theta_i \wedge \varphi(c)$.
   If $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is not consistent, do nothing. ($\theta_{i+1} = \theta_i$.)

Want T_∞ to (1) be complete, (2) have witnesses, (3) omit p

We decide θ_{i+1} depending on the strength of $T \cup \{\theta_i\}$.

1. (Stage $i = 3k + 1$: ensuring completeness)
   Decide which of σ_k, $\neg\sigma_k$ to put in T_∞:
   If $i + 1 = 3k + 1$, and $T \cup \{\theta_i\} \models \sigma_k$, then let $\theta_{i+1} = \theta_i \wedge \sigma_k$, else let $\theta_{i+1} = \theta_i \wedge \neg\sigma_k$.

2. (Stage $i = 3k + 2$: ensuring witnesses)
   Assume that σ_k is $(\exists x)\varphi(x)$ where $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is consistent.
   Choose a constant c that does not appear in $T \cup \{\theta_i\}$.
   Let $\theta_{i+1} = \theta_i \wedge \varphi(c)$.
   If $T \cup \{\theta_i, (\exists x)\varphi(x)\}$ is not consistent, do nothing. ($\theta_{i+1} = \theta_i$.)

Want T_∞ to omit p

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula,

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \to \psi(\mathbf{x}))$.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \to \psi(\mathbf{x}))$.
   Hence, some model M of T has a tuple $\mathbf{s}$ realizing $\delta(\mathbf{x})$ that does not realize $\psi(\mathbf{x})$.

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \to \psi(\mathbf{x}))$.
   Hence, some model M of T has a tuple $\mathbf{s}$ realizing $\delta(\mathbf{x})$ that does not realize $\psi(\mathbf{x})$.
   Interpret $\mathbf{c}_k = \mathbf{s}$.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$.
   Hence, some model M of T has a tuple $\mathbf{s}$ realizing $\delta(\mathbf{x})$ that does not realize $\psi(\mathbf{x})$.
   Interpret $\mathbf{c}_k = \mathbf{s}$. $M \models \delta(\mathbf{c}_k) = (\exists \mathbf{y})\gamma(\mathbf{c}_k, \mathbf{y})$, so there is a choice for $\mathbf{d}$ so that $M \models \gamma(\mathbf{c}_k, \mathbf{d}) = \theta_i$.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$.
   Hence, some model M of T has a tuple $\mathbf{s}$ realizing $\delta(\mathbf{x})$ that does not realize $\psi(\mathbf{x})$.
   Interpret $\mathbf{c}_k = \mathbf{s}$. $M \models \delta(\mathbf{c}_k) = (\exists \mathbf{y})\gamma(\mathbf{c}_k, \mathbf{y})$, so there is a choice for $\mathbf{d}$ so that $M \models \gamma(\mathbf{c}_k, \mathbf{d}) = \theta_i$.
   $M_{\mathbf{cd}}$ is a model of $T \cup \{\theta_i\}$ in which $M \not\models \psi(\mathbf{c}_k)$.

Want T_∞ to omit p

1. (Stage $i = 3k + 3$: ensuring type omission)
   Let $\mathbf{c}_k$ be the next n-tuple to be considered.
   Write θ_i so that it is a statement about $\mathbf{c}_k$:
   Let $\gamma(x_1, \ldots, x_n, \mathbf{y})$ be the formula obtained from sentence θ_i by (i) replacing each $c_{k,i}$ with x_i and (ii) replacing every other constant d_j from θ_i with some variable y_j. So θ_i is $\gamma(\mathbf{c}_k, \mathbf{d})$.
   Then $\delta(\mathbf{x}) = (\exists \mathbf{y})\gamma(\mathbf{x}, \mathbf{y})$ is an L-formula, which cannot support p.
   There must exist $\psi(\mathbf{x}) \in p$ such that $T \not\models (\forall \mathbf{x})(\delta(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$.
   Hence, some model M of T has a tuple $\mathbf{s}$ realizing $\delta(\mathbf{x})$ that does not realize $\psi(\mathbf{x})$.
   Interpret $\mathbf{c}_k = \mathbf{s}$. $M \models \delta(\mathbf{c}_k) = (\exists \mathbf{y})\gamma(\mathbf{c}_k, \mathbf{y})$, so there is a choice for $\mathbf{d}$ so that $M \models \gamma(\mathbf{c}_k, \mathbf{d}) = \theta_i$.
   $M_{\mathbf{cd}}$ is a model of $T \cup \{\theta_i\}$ in which $M \not\models \psi(\mathbf{c}_k)$.
   Let $\theta_{i+1} = \theta_i \wedge \neg\psi(\mathbf{c}_k)$.

Final assembly

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}.$

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\Box$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\Box$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently (e.g. $c_i \neq c_j$, $c_i \neq d_j$, $d_i \neq d_j$).

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently (e.g. $c_i \neq c_j$, $c_i \neq d_j$, $d_i \neq d_j$).
Let $p(x)$ be the partial 1-type consisting of all $(x \neq c_i)$.

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently (e.g. $c_i \neq c_j$, $c_i \neq d_j$, $d_i \neq d_j$).
Let $p(x)$ be the partial 1-type consisting of all $(x \neq c_i)$.
p is not supported, but cannot be omitted.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently (e.g. $c_i \neq c_j$, $c_i \neq d_j$, $d_i \neq d_j$).
Let $p(x)$ be the partial 1-type consisting of all $(x \neq c_i)$.
p is not supported, but cannot be omitted.

Final assembly

$T_\infty = T \cup \{\theta_i | i \in \omega\}$.
T is a Henkin theory in which no tuple of constants realizes p.
The Henkin model will not realize p. $\square$

Remark.
Theorem is false as stated for uncountable languages.

The Standard Counterexample. Let L be a language with constants only, $\{c_i \mid i \in \omega\} \cup \{d_j \mid j \in \omega_1\}$.
Let T be the theory axiomatized by sentences saying that all constants interpret differently (e.g. $c_i \neq c_j$, $c_i \neq d_j$, $d_i \neq d_j$).
Let $p(x)$ be the partial 1-type consisting of all $(x \neq c_i)$.
p is not supported, but cannot be omitted.

A carefully worded restatement of the theorem is true for uncountable languages.